report

CR–182

Copy 1 of 5 Copies

Documentation of the Tactical Vehicle Fleet Simulation Model

VOLUME II – ANALYST/PROGRAMMER MANUAL

PART A – Narrative

By

Paul E. Hudak
Lawrence D. Koenig
George O. Swanson

Prepared for:

Department of the Army
U.S. Army Logistics Center
Fort Lee, Virginia

Contract No. DAAG38-77-R-9055

May 1977

GENERAL RESEARCH CORPORATION
WESTGATE RESEARCH PARK, McLEAN, VIRGINIA 22101

CONTENTS

Chapter 1
INTRODUCTION

GENERAL DESCRIPTION OF THE MODEL

The Tactical Vehicle Fleet Simulation (TVFS) Model is a computerized stochastic representation of vehicle fleet operations. The model accepts descriptions of vehicles comprising a fleet as to capabilities and operational characteristics. Then, based on the generation of mission demands, it dispatches, loads, transits, unloads, returns, and maintains the vehicles over time. The model reports the overall utilization of the fleet and measures vehicle performance in terms of dispatch delays and in terms of the times required to complete requested missions.

On a more generalized basis, the TVFS model can be described as a discrete event queueing and service model. It generates events, such as missions and maintenance actions, according to prescribed statistical distributions. It then keeps account of the time required to service each of these events with respect both to the type of event and to the type of server (i.e., vehicle). Results are displayed in terms of explicit frequency distributions of the delay and service times, as well as comprehensive summary statistics.

HISTORY OF MODEL DEVELOPMENT

The TVFS model has gone through four basic stages of development since its inception in 1965. Initially, the modeling methodology was developed to support the analysis of helicopter operations. (Reference 1.) The basic model logic was subsequently part of an integrated methodology, designated the Candidates Family Methodology. This analytical framework was composed of the simulation model and two other interfacing models, a cost model and an optimization model. (Reference 2.) Still later, the model itself was extensively modified and used to study opportunities for vehicle pooling. (References 3 and 4.) Most recently, the

model was adapted to accept vehicle travel time data generated by the US Army, Corps of Engineers Waterways Experiment Station. This latest version of the model was used to assess the comparative performance of alternative fleets composed of standard and high mobility vehicles. (Reference 5.)

It is this latest version of the TVFS model that is primarily documented here. However, to facilitate application of the model to problems similar to previous studies, most of the earlier specialized logic has been retained in the model code and has also been documented here. The portions of the model not exercised for the high mobility study are appropriately noted throughout this document.

CURRENT SCOPE AND CAPABILITIES

The TVFS model, as currently written, can be used to evaluate the effectiveness of a given fleet of vehicles with respect to a given set of missions generated over time. The model allows the evaluation of alternative compositions of the vehicle fleet, the operational procedures for assigning vehicles to missions, the effect of different mission demands on fleet performance, and the effect of such secondary factors as maintenance procedures and loading/unloading capabilities on performance of missions. The model provides a logical framework for synthesizing detailed data on the supply support aspects of combat scenarios, vehicle characteristics, and mission demands. The model operates with several types of vehicles simultaneously, simulating the performance of a given fleet on a given mission pattern over a specified period of time. Alternative fleet sizes and compositions are run successively to compare their performance.

In its latest application, the TVFS model was used to quantify the comparative performance of vehicle fleets that were composed of different mobility capable vehicles and that provided forward supply support to units of a brigade. Four basic mission types were considered in the analysis: general cargo resupply, ammunition resupply, bulk petroleum-lubricants-oil (POL) resupply, and salvage/retrieval missions. A single simulation run was characteristized by a combination of scenario, combat

posture, weather, route, mission types and vehicle fleet. A complete discussion is presented in later sections of how these aspects are represented by data inputs to the model. Other applications of the model to similarly represented problems will undoubtedly be evident to the reader as that discussion proceeds. The TVFS logic is generalized enough to allow its usage in evaluating a broad range of vehicle fleet performance problems besides those previously modeled.

ORGANIZATION OF DOCUMENTATION

The documentation of the TVFS model is organized into three separate volumes, corresponding to three different reader audiences. Volume I is an Executive Summary and is oriented to senior managers. It provides an overview and general description of what the model is and what it does. Technical details and instructions on how to use the model are excluded from this volume. The Executive Summary includes a discussion of the capabilities and limitations of the model, the basic assumptions underlying its logic, input requirements, and available outputs. A summary of the resources required for a representative application of the model to a study is also included in Volume I.

Volume II of the documentation is aimed at the programmer-analyst who will be responsible for actually running the model. In addition to the general descriptive matter of Volume I, this volume contains detailed information about the model's structure, subroutine logic, variable definition, input data requirements, and output formats and options. It includes system, macro, and micro flowcharts, as well as complete source code (FORTRAN) listings. Design assumptions, rationale, and mathematical representations are also presented in Volume II.

Volume III is a Planner-User Manual and describes the model and its use from a functional point of view. It contains the same general descriptive introduction of Volumes I and II. Thereafter, it covers data input in extensive detail. A complete set of data for a sample problem is presented and is later carried forward to illustrate typical output reports and related manual analyses. Volume III also contains specific instructions for operating and controlling the model, as well as computer related requirements such as core storage and model run times.

REFERENCES

1. Research Analysis Corporation, "Simulation Model for Vehicle Operations," RAC-T-475, February 1966.

2. _____, "The Candidate Families Methodology: Simulation, Cost, and Optimization Models with User's Manual" (Volume I, Part II: Simulation Model), RAC-D8(TP-395), May 1971.

3. _____, "Analysis of Opportunities for the Reduction of Tactical Vehicle Requirements through Pooling," RAC-TP-420, April 1971.

4. _____, "Tactical Vehicle Pooling in the Corps/Army Service Area," RAC-R-143, July 1972.

5. General Research Corporation, "Special Analysis of High Mobility Tactical Vehicles by Application of the Tactical Vehicle Fleet Simulation Model," OAD-CR-125, October 1975.

Chapter 2

OVERVIEW OF MODEL STRUCTURE

GENERAL DESIGN

The simulation model is designed to evaluate the performance of a given fleet of vehicles for a given set of mission requests. It operates by stepping through a series of missions, assigning vehicles to each mission, while keeping track of total operating time for each vehicle, delays due to unavailability of vehicles, and various other measures of performance. It also withdraws vehicles at appropriate (random and/or deterministic) times, for scheduled and for random periods, according to a specified pattern. Various options are available to the user, as explained in later sections.

The model consists of 74 subroutines, seven functions and a main routine. All programs are coded in FORTRAN V. Extensive common blocks allow passage of variable data needed in particular subroutines, with a minimum number of calling parameters. All subroutines are linked to the main program via calls from either the main routine or another subroutine. The sequence of calling statements generally follows the flow of data from input card form, through a validation phase, to simulation initialization.

FUNCTIONAL LOGIC

Figure 1 is a schematic of major simulation functions and shows the activities that occur during one time increment of simulated fleet operations. The simulation can run for any given period of operating days and each day is subdivided into time increments. The increments chosen are of such magnitude that time accuracy of simulation output is achieved and time assumptions for stochastic input variables are met. For instance, accuracy of a specification that certain unscheduled missions occur according to a Poisson distribution depends on time intervals being small enough

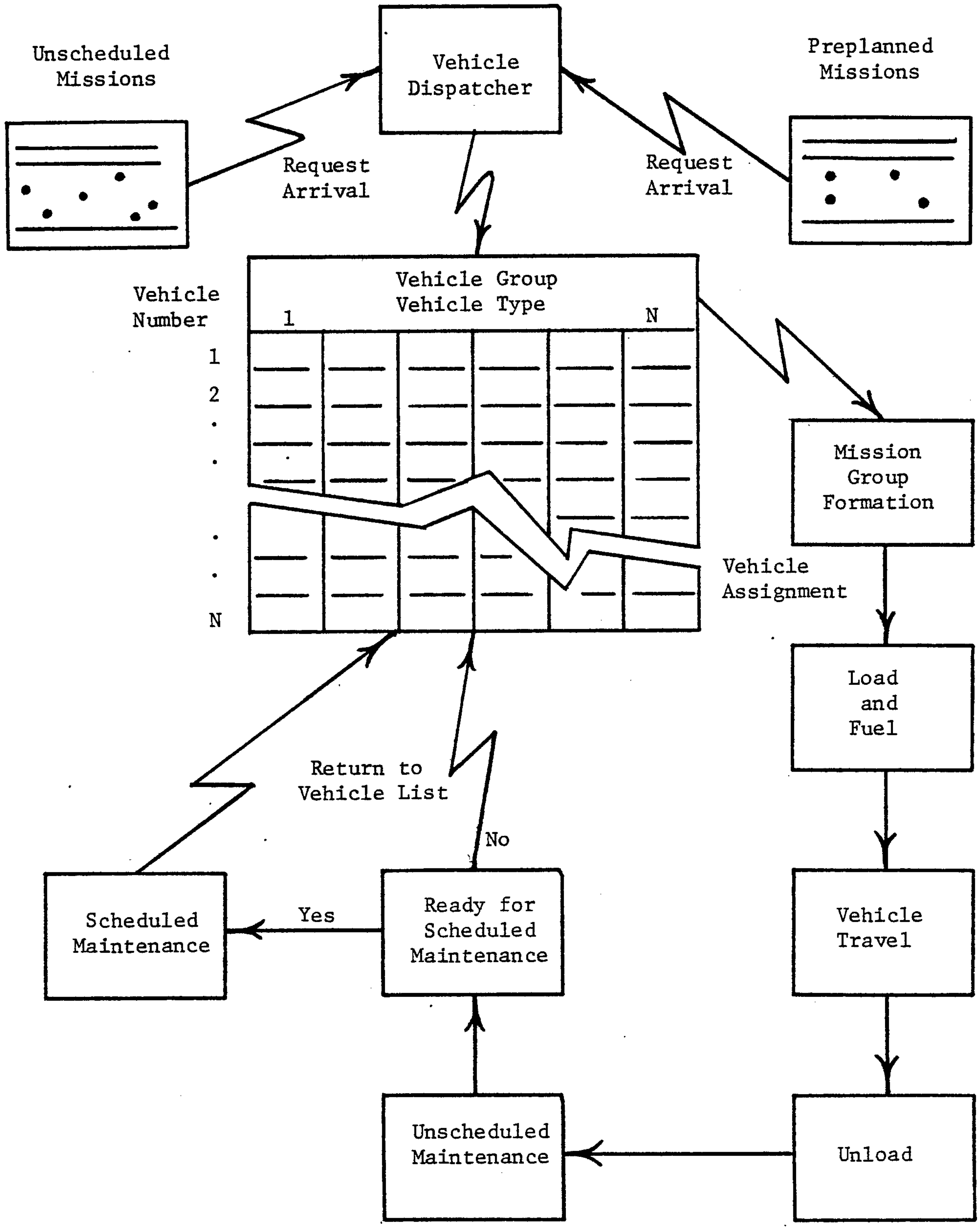

Figure 1—TVFS Simulation Schematic

that the probability of mission occurrence in any one time interval is relatively small.

Figure 1 reveals that the simulation handles two types of missions, i.e., preplanned or unscheduled. The request of these missions initiates simulation activity. Preplanned missions are submitted to the vehicle dispatcher in a specified daily schedule or for the entire simulation duration. Alternatively, unscheduled missions are generated by random demands for vehicle services during each operating day by the organizations requiring resupply.

Upon receipt of a mission request, the vehicle dispatcher routine (see Figure 2) notes the priority of the requested mission and evaluates this priority in comparison with the priority of all other mission requests that occur during the same time interval. Missions are serviced by the vehicle dispatcher in priority order during any specific time increment. All priority 1 missions are dispatched first, then priority 2, and so on. Within groups of missions of the same priority, mission service or vehicle dispatch is on a first-come, first-served basis.

When a requested mission is ready to be serviced by the vehicle dispatch routine, the dispatcher notes the vehicle group from which the vehicle to perform the mission is to be taken and notes the vehicle types within this group that are allowed to perform the mission. Among these vehicle types, the vehicle dispatcher scans the vehicle fleet for vehicles available to perform the mission. This scanning is done in the order of vehicle preference specified for the type of mission requested. That is, if more than one vehicle type is allowed to perform the mission requested, the dispatcher will search for vehicles of first preference, then second preference, and so on.

Proceeding in this manner, the vehicle dispatcher forms one or more groups of vehicles (if more than one vehicle is required) to perform the mission. The formation of vehicle groups proceeds under one of two sets of rules. If homogenous group service is specified, vehicle groups will be formed from only one vehicle type. If combination service is specified, the dispatcher forms a group of vehicles from the vehicle types most available.

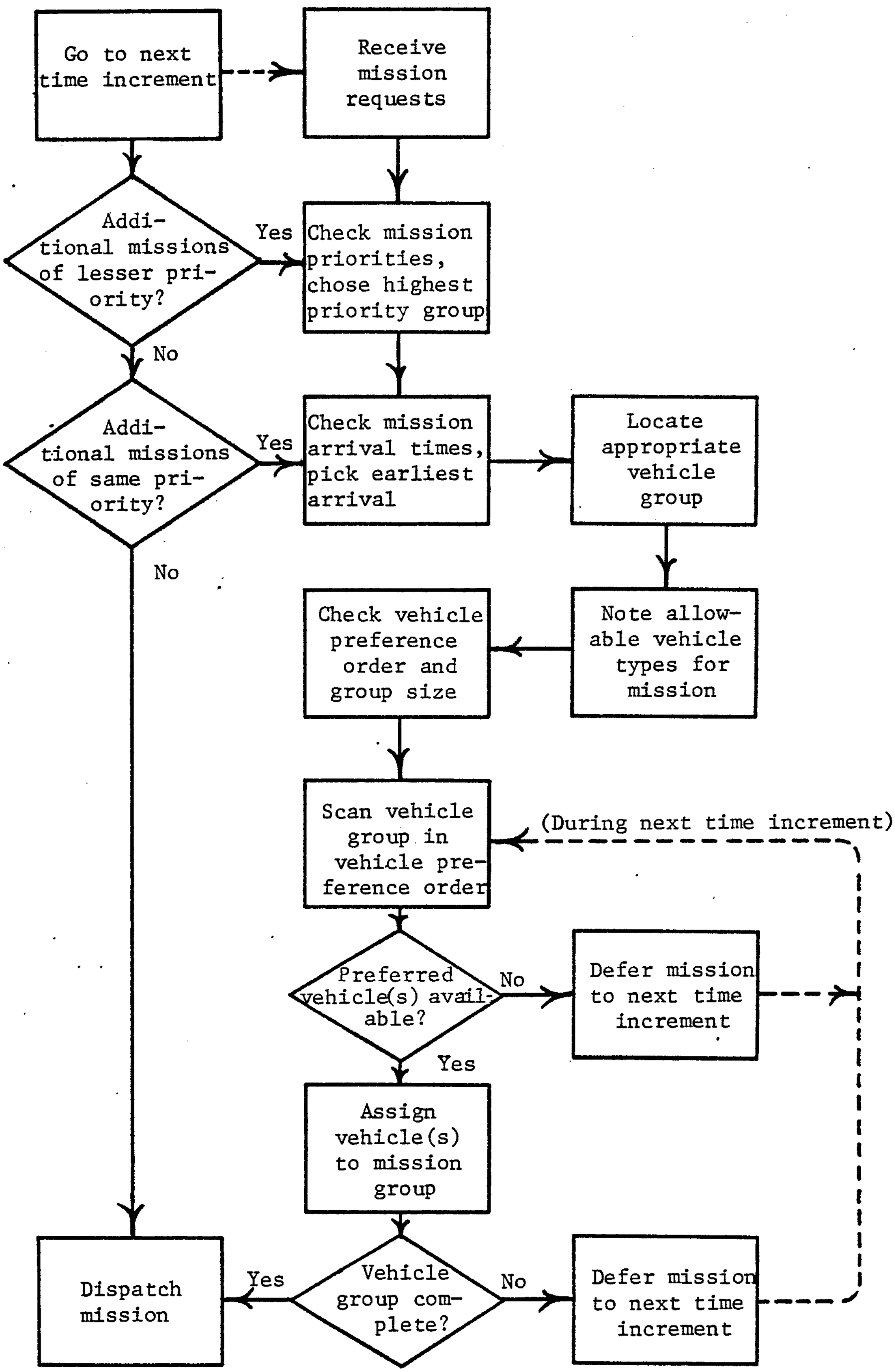

Figure 2—Mission Dispatch Routine

If in scanning the vehicle fleets the dispatcher finds no vehicles of the proper type to assign to the mission, the mission is deferred until the beginning of the next time increment, when the whole process begins again. If some but not all of the proper vehicle types are located, the vehicles found are placed in a group forming area; in the next time increment (or succeeding increments), the search for vehicles is continued until the required number are found. If the proper vehicles are found during the initial search, the vehicle group is dispatched immediately to perform the mission.

Upon initiation of mission dispatch, the vehicles are loaded and fueled and then proceed to the mission destination. They then unload and return as shown in Figure 1. Before the vehicles are returned to an availability status, however, they are checked by the maintenance routines for the necessity of scheduled or unscheduled maintenance. The need for unscheduled maintenance is determined by drawing a random number and comparing it with a distribution of unscheduled maintenance occurrences generated from basic input data on vehicle reliability and maintainability. If by this process it happens that the vehicle draws unscheduled maintenance, another similar random process generates the appropriate unscheduled maintenance time. The form of both these unscheduled maintenance distributions and their necessary parameters are part of the simulation input.

Upon completion of unscheduled maintenance or if no unscheduled maintenance is assessed against the vehicle, a check of each vehicle's maintenance history is made to see whether the vehicle has accumulated enough operating hours to require scheduled maintenance before the vehicle returns to the available fleet. If so, a number of scheduled maintenance hours (dependent upon vehicle type) are allowed to elapse before the vehicle is allowed to perform further missions. In checking the need for scheduled maintenance, the model employs a threshold such that if the vehicle is not due for scheduled maintenance but would be due before completion of the next assigned mission, the vehicle goes to maintenance before returning to its vehicle group. This maintenance threshold is based upon average mission durations.

Following the maintenance routines, the vehicle is returned to the appropriate group. During this cycle of vehicle operations, other vehicles are being dispatched and perform requested missions as the time increments of simulation move forward in time. As the simulation proceeds, detailed statistics for all vehicles and missions are accumulated and become the bases for simulation output parameters, i.e., data that serve as the bases for analyses of the simulations performed.

INPUT DATA

Operation of the TVFS model requires three basic types of input data: (1) mission definitions and demand schedules, (2) vehicle types and vehicle fleet characteristics, and (3) simulation control parameters. Mission related data include organizational unit locations, kinds of missions, relative mission priorities, mission payloads, travel times between unit locations, vehicle-mission assignment preferences, and unit load/unload rates. Vehicle related data include vehicle capacities, speeds (optional), maintenance history, and maintenance requirements. Simulation control parameters include duration of simulation, mission generation and performance periods, and mission discipline descriptors. Detailed specifications of input data variables are documented in Volume II, Chapter 4; a complete input deck for a sample problem is presented in Volume III, Chapter 6.

MODEL OUTPUT

The TVFS model produces several kinds of output about mission execution and vehicle fleet performance. In particular, it generates the following reports for each mission type:

- Number of missions requested -- in total and by day.
- Number of missions completed -- in total, by day, and within specified periods.
- Number of missions not met.
- Number of missions delayed -- in total, by day, and by length of delay.

For each general type of vehicle, reports are produced that show:

- Number of vehicles utilized each simulation day.
- Time utilization by major activity, i.e., transit, loading, maintenance, etc.

● Intermediate simulation status of vehicles with respect to availability for and performance of missions.

All output reports and the methods for controlling their generation are fully documented in Volume II, Chapter 5. Also, Volume III contains example output reports and an explanation of their interpretation for the sample problem.

BASIC ASSUMPTIONS

There are two sets of general assumptions which pertain to the model. The first set relates directly to the TVFS model itself and deals with how the model was programmed. The second set concerns how the model was applied to the latest study.

General Assumptions

This set of assumptions deals basically with the type of statistical distributions used, order of processing events and general tie-breaking rules. The logic of the TVFS model assumes that:

1. Mission requests can be adequately represented as a compound Poisson process, or equivalently, as a simple Poisson process for each type of mission that operates independently. This determines how missions are generated during a specified request period.

2. Priority classes can be specified such that mission requests are honored on a first-come, first-served basis within each class. For example, general cargo and ammunition missions are modeled in the same simulation run. Ammunition missions are given first priority. Within this priority class, all ammunition missions are dispatched on a first-come, first-served basis. No other distinction in priority is made among ammunition missions.

3. Downtimes can be represented as a combination of scheduled maintenance times based on accumulated operating hours, randomly distributed unscheduled maintenance times required after some randomly chosen missions, and other non-transit times that correspond to load/unload/checkout/crew change times and that are fixed for given vehicle and mission combinations. With these assumptions the model has been designed to use pseudo-random numbers to supply randomness, and all other required information can be furnished as input.

Assumptions Relating to the Latest Model Application

Other assumptions are implied by the logic of the TVFS model and/or the way in which problem data are structured for input to the model. This second set of assumptions relates specifically to the latest application of the model and allows the structuring of the various types of available data into a form amenable to representation by the general logic of the model. It is assumed that:

1. The demands for resupply tonnages generated by forward combat units are serviced by vehicles organic to those units and not by vehicles of service support units in the rear support area.

2. Certain types of missions are performed exclusively by a dedicated and identifiable set of vehicles organic to forward combat units. Vehicles are assignable to independent groups of missions: cargo/ammunition, bulk POL, and salvage/retrieval. For example, one set of vehicles is allowed to perform POL missions and another set of vehicles is allowed only to perform retrieval missions. The model does not account for the case where a tanker vehicle is used to tow another vehicle and thus be unavailable to conduct its designated mission.

3. Vehicle attrition does not vary across fleets. The TVFS model does not account for vehicle attrition due to combat actions, accidents, irreparable breakdown, etc. To the extent that attrition causes congestion in the vehicle resupply system, mission completion times and vehicle utilization statistics will be misstated. However, this effect is judged to be small if one also judges that most attrited vehicles are replaced from rear area reserve pools. For purposes of comparing the performance of alternate fleets, the attrition effects are considered to fall in the latter category, i.e., the losses are offset by replacement.

4. Vehicles are not tagged by their owning units. Thus, the model assumes that vehicles of one particular unit may be available to perform resupply to other units and that this "swapping" cancels out over the full simulation period.

5. The daily resupply tonnage to a particular unit can be carried by whatever allowable vehicle types are available when the mission occurs during an operating day. That is, nonhomogeneous groups of vehicles may perform a resupply mission to a particular unit.

6. Forward resupply missions are performed by vehicles traveling directly from some specified rear area supply point to the forward requesting unit without transloadings or other delays at intermediate points along the route.

Other aspects of the analysis deriving from the nature of data provided for the HIMO simulations and from simplification inherent in the model should be considered by the reader in evaluating applicability of the results. These aspects include the following:

1. Unit relocations from one day to the next in a scenario were not modeled.

2. The conduct of special tactical missions by either standard or high mobility vehicles was not modeled.

3. Only two fleet mixes were examined as alternatives to the existing TOE fleet. The alternatives were extreme compositions, i.e., all-standard and all-high mobility vehicles.

4. The TVFS model is not an inventory model. That is, it does not keep track of stock levels at either forward unit locations or rear area supply points. Consequently, its output measures of fleet performance may be difficult for logisticians to interpret.

5. Data provided on vehicle travel times and routes included only one off-road route option. This option was, in fact, only partially off-road. More severe off-road cases were not modeled.

Chapter 3
DETAILED STRUCTURE AND LOGIC

This chapter discusses the general structure and the logical flow of data through the model, as these pertain to the current application. The following sections are designed to give the programmer-analyst a knowledge of the logical flow through the various subroutines, thereby allowing an individual with a knowledge of FORTRAN to follow simulated events in a predictable manner. These sections will provide a more comprehensive look at the structure and logic of functional groups of routines as well as individual subroutines, in order to demonstrate the sequence of simulation and the hierarchy of the model logic.

PROGRAM STRUCTURE

Many of the subroutines within the TVFS model perform similar functions and many are unique to a particular event or circumstance within the course of execution. A detailed study of the subroutines will verify that very few routines are used strictly for input or output or initialization, but rather that a subroutine may cross several functional areas.

Table 1 represents the basic function of each of the TVFS subroutines. The categorization of a particular subroutine under any one function heading should not be construed as an indication that none of the other functions is performed by the particular subroutine, but that this is the primary purpose of the subroutine.

DEFINITION OF VARIABLES

The TVFS model is designed to allow passage of control between various subroutines with a minimum of calling arguments by use of COMMON blocks. While this is convenient in terms of simplicity of code, it does require extensive COMMON blocks to carry necessary information between the appropriate routines.

Table 1

SUMMARY LISTING OF SUBROUTINES GROUPED BY FUNCTION

Input	STORØ2	PPMSNS
CARDS	STORØ4	PRVHAV
CARDXX	STOR11	QUESCH
NEWSRCS	STOR12	RANDØ
POSTURE	STOR13	RANDØA
RDMSDL	STOR25	RAND1
RDPOOL	STOR43	RAND2
RDSRCS	ST1325	RAND3
RDTTM	Simulation	SCHMSN
Output	ASMNT1	SRTPER
DAYOUT	ASMNT3	SRTPRF
OUTPT1	ASSMNT	STORPP
OUTPT2	CALCDST	STVMPF
OUTPT3	CNTRL2	TYPRATE
OUTPT4	CNTRL5	VEHSUM
OUTPT5	DAYREQ	VHMSPRF
OUTPT6	DELAST	System Clock
PRNTIN	DELETE	and Timing
PRNTN1	GRPSUM	ADJTME
PRNTN2	IGPSRV	ASMNT1A
PRNTQUE	INSERT	ASMNT1B
WRMSDL	MINVEHS	ASMNT2
Initialization	MOVPOOL	CNTRLØ
CLRQUE	MOVSRC	CNTRL3
CNVXY	MOVSRCS	CNTRL4
CTEDPPM	MSNCTGP	NEXTME
FSTOR1	MSNCTPR	MNTVEH
FSTOR2	MSPRCT	MSNPCT
ISTOR1	NEXTIN	TMINCR
ISTOR2	NXAVPAC	
SETUP1	NXTREQ	

The model uses six labeled common blocks plus several small unlabeled common blocks for all data transfer between routines. Within subroutines, local variables are used, i.e., variables which are introduced and initialized at the beginning of a routine, and whose values are lost or transferred to common when an exit from the routine occurs. Many of these local variables are used in different contexts within various routines and therefore cannot be defined, except as they may be used within a specific routine. Most of the variables carried within common blocks are arrays containing data related to input values or values of data generated as a result of the simulation. Table 2 is a list of all variables used within the simulation, with the exception of some local variables used as indices, data exception flags and pointers. This table also lists any common block with which a variable may be associated or the subroutines for which their use is restricted. (Variable definition may differ from time to time during the simulation as in the case of a variable which might represent an input value upon initialization of the program and be summed with another variable as execution continues.)

SYSTEM FLOW DESCRIPTION

Figure 3 represents the system flow diagram of the TVFS model. Each of the major processing blocks is discussed in general terms below. More detailed descriptions of the processing blocks appear in later sections of this volume.

Figure 3 shows that after initialization, vehicle and mission characteristics are read as input data and checked for completeness. Existing errors or inconsistencies are printed with error messages. Maintenance times are calculated, and vehicle workload data are printed. Simulation arrays and variables are initialized and vehicle age is established. The mission request queue is then cleared and loaded with the requests of the day. Arrays and variables for daily totals are initialized and the calendar is incremented. Time of day is then determined by the soonest available vehicle.

If it is not the end of the day and the request queue is not empty, fulfillment of a mission request begins. The soonest available preferred vehicle type is selected, and a vehicle of this type is assigned to the

Table 2

DEFINITION OF VARIABLES

Common	Variable	Description
DAYTIM	ACHRDY	Vehicle operating hours per day if the only vehicle type
DAYTIM	ACMIN	Minimum number of vehicle required
VEHRY1	ADTIME(I)	Time beyond last day for vehicles of type I
common	AMHFH(I,J)	Maintenance hours per operating hour for vehicle type J on mission I
DAYTIM	AMMO	Index to priority
	AVCMP	Average delay completed missions (hours)
	AVDLD	Average delay delayed missions (hours)
	AVDLTM	Average delay time (minutes) per mission type
	AVDYDY	Average delay for only delayed
OUTSUM	AVEHUS	Number of vehicles of all types used per day
	AVGSPD(I)	Average speed for vehicle I
ACØ3	AVMNTME	Average maintenance time (hours)
OUTSUM	AVNAP	Average number of vehicles of a type used per day
	AVREQ	Demand rate -- missions per day
	AVSORT	Average number of missions completed per day of a priority
ACØ3	AVOSRTM	Average sortie time (hours)
	AVSTDY	Average delay for all missions of a priority
DAYTIM	BLKFUEL	Index to priority
common	BNDSDM(I,J)	Ith time interval for delayed priority J missions
	CC1	First digit of card ID
	CC2	Second digit of card ID
common	CFP(I,J)	Cumulative probability for mission type I of priority J
REQRYS	CMPCTG	Percent of completed missions per priority for which to calculate delay time
REQRYS	CMPTME	Delay time for percent of completed missions (CMPCTG)
REQRYS	CTSTDY(I)	Total missions completed today of priority I
REQRYS	CTSTRN(I)	Total missions completed to date of priority I

Table 2 (continued)

Common	Variable	Description
DAYTIM	DAOUSW	Print option switch Ø = No daily report 1 = One page daily report (full) 2 = One page daily report (abbreviated) 3 = Tabular daily report (one line per day of operation)
	DAYDEL	Delay time today (hours)
REQRYS	DAYDLY(I)	Daily cumulative mission delay time by priority
DAYTIM	DAYMOYR	Date of simulation run
A1Ø	DEL	Mission delay time
	DELAY	Total delay time to date (hours)
VEHRY1	DELGRP(I)	Reciprocal of vehicle type I group size for current mission
common	DELTM(I,J)	Delay time to date for type I missions of priority J
common	DELTM2(I,J)	Square of delay time to date for type I missions of priority J
	DEVNAP	Standard deviation of average number of vehicles used per day of a type
	DEVSOR	Standard deviation of missions completed per day
JOBTMS	DISCAL	Distance - common ammunition links
JOBTMS	DISCCL	Distance - common cargo links
JOBTMS	DIST(I).	Distances - supply point - SRC unit(I)
	DISTOT	Total distance across links
	DLTM	Delay time
	DLTM2	Delay time
	DNTIM	Total down time (minutes)
common	DOPHRS(I,J)	Daily operating (travel) time for vehicle I of type J
POOLSP	DSPLSP	Distance pool supply point moves
	DST	Distance of move
common	DURSCH(I,J)	Time a type J vehicle spends in step I of scheduled maintenance
A11	DX	Distance from old unit location to new unit location
	E	(=1.ØE-5)
VEHRY1	ENTGMO(I)	Expected maintenance time (given maintenance occurs) for a type I vehicle

Table 2 (continued)

Common	Variable	Description
	EPSILN	(=1.ØE-1)
	FASSB	Time assignment begins
	FASSE	Time assignment ends
	FBM	=FEBAMVE(I)
DAYTIM	FDAY	Length of day in minutes
	FDAYB	Time day beings
	FDAYE	Time day ends
DAYTIM	FEBACYC	Advance cycle
common	FEBAMVE(I)	Unit rate of advance for day I
REQRYS	FM(I)	Unscheduled mission frequency per time increment for priority I missions
	FMVT	New minimum number of vehicles
	FMSNLD	Average mission load time (hours)
	FMSNULD	Average mission unload time (hours)
	FMST	Time vehicle travelled distance of mission
	FN	Missions completed
common	FNP	Input 16-word buffer area
CARD	FNPUT(I)	Floating point copy of INPUT(I)
	FREQB	Time requesting begins
	FREQE	Time requesting ends
	FSOR	Mission time
INTIME	F1	Total time increments in the time period
	GFWT	Total group-forming wait times
common	GNP	Input two-word buffer area
	GP	Group size
LIMITS	GPCOMP	Requirement for mission group completion (=1.Ø)
common	GPFMWT(I,J)	Group-forming waiting time for vehicle I of type J
OUTSUM	GRPFWT	Group-forming waiting time for all vehicles
DAYTIM	GRPSAV	Used in conjunction with transfer of elements of larger arrays
VEHRY1	GRPSIZ(I)	Group completion counter for type I vehicles
common	HRSCH(I,J)	Time until a type J vehicle enters step I of scheduled maintenance

Table 2 (continued)

Common	Variable	Description
REQRYS	HSTMIN(I)	Width of frequency interval for delayed missions of priority I
VEHRY1	IACND	Number of vehicles needed to complete a group for the current request
VEHRY1	IACT(I)	Count of vehicles in each mission group for the current request
DAYTIM	IASSB	Time assignment of vehicles begins each day
DAYTIM	IASSE	Time assignment of vehicles ends each day
common	IAVLT(I,J)	Encoded time assigned, vehicle number, time available
DAYTIM	IAVT	Simulation time (time increment)
VEHRY1	IAVWT	Time when vehicle was first assigned to a group
DAYTIM	ICARRY	Mission servicing procedure option
	ID	Link ID
	IDAYS	Preplanned request day
DAYTIM	IDAYB	Time working day begins
DAYTIM	IDAYE	Time working day ends
common	IDMSN(I,J)	Daily total of delayed type I missions of priority J
common	IDTREQ(I)	Storage for scheduled missions
DAYTIM	IERROR	Card error indicator (=1)
DAYTIM	IEVEN	Vehicle assignment option
	IFIND	Index in a circular scheduled maintenance table
	IFINDX	Index in a circular scheduled maintenance table
	ILAB	Label (='AVG SPD')
INTIME	INASSB	Time assignment of vehicles begins each day
INTIME	INASSE	Time assignment of vehicles ends each day
INTIME	INDAYB	Time working day begins
INTIME	INDAYE	Time working day ends
DAYTIM	INPPM	Indicator for preplanned mission requests ($\geq$ 1)
DAYTIM	INPT1	Input unit for CARDXX and preplanned mission requests
CARD	INPUT(I)	Input area for parameter cards
INTIME	INREQB	Time requesting of missions begins each day

Table 2 (continued)

Common	Variable	Description
INTIME	INREQE	Time requesting of missions ends each day
CARD	INTYPE(J)	Data type indicators Ø = Input(J) is integer 1 = Input(J) is floating point
REQRYS	IOUT	See OUT1
	IPR	Priority
DAYTIM	IPRINT	Unit designation (=6) for system output device (PRINTER)
DAYTIM	IPRNT1	Options to print (=1) step by step simulation results
DAYTIM	IPRNT2	
DAYTIM	IPRNT3	
DAYTIM	IPRNT4	
DAYTIM	IPRNT5	
DAYTIM	IPRNT6	
DAYTIM	IPRNT7	
DAYTIM	IPRNT8	
DAYTIM	IPUNCH	Unit designation (=7) for punched results of delayed missions
	IR	Mission delay read counter
A1Ø	IREC	Mission delay record number
DAYTIM	IREQB	Time requesting of missions begins each day
DAYTIM	IREQE	Time requesting of missions ends each day
	IRQ	Request
	ISTAR	Vehicle assignment starting point indicator
common	ITMSN(I,J)	Total type I missions of priority J completed
REQRYS	IRSRQ(I)	Total priority I missions requested
DAYTIM	ITYPE	Number of vehicle types this run
DAYTIM	IU	(=NUNITS)
	JABT	Indicator for correspondence between job-travel time SRCS and the SRC of the mission-unit data set
JOBTMS	JBSRC	Job SRC
common	JNP	Input 13-word buffer area
DAYTIM	J1	Time constant (J2 + KLOCK)

Table 2 (continued)

Common	Variable	Description
DAYTIM	J2	Full days of mission delay (not including today)
DAYTIM	J3	Calendar (time increments) - full days only
DAYTIM	KLKDAY	Day of simulation
DAYTIM	KLOCK	Time of day (time increments)
REQRYS	KNDREQ	Type of unscheduled mission generated and to be put in queue
	KPRI	Decoded priority
CARD	KQTY	Number of parameter card fields
	KREQ	Decoded request
REQRYS	KSTREQ(I)	Number of missions requested per day for priority I
	KTME	Decoded time of request
	LEFT	Requests remain for this day indicator
ACØ3	LGSTMIN	Minimum vehicles corresponding to largest work-load
DAYTIM	LIMIT	Number of days for current run
	LNCT	Line-counter
DAYTIM	LOADTM	Loading time
REQVAR	LSTAVL	Last available location in the mission request queue
	MAP1	First map code
	MAP2	Second map code
LIMITS	MAXPRS	Maximum number of mission priorities acceptable
A1Ø	MAXREC	Maximum number of delayed mission punched records
DAYTIM	MAXSRC	Maximum number of SRC units acceptable
	MAXVH	Maximum vehicle number
MCTØ1	MCTHIST(I,J)	Frequency of mission completion times in intervals of MCTINT minutes
MCTØ1	MCTINT	Mission completion time interval
MCTØ1	MCTMXI	Maximum mission completion time interval
MCTØ1	MCTMXM	Maximum number of priorities for which mission completion time is calculated
DAYTIM	MCTRTT(I,J)	Total mission completion time (in minutes)
	MDTD	Missions delayed to date

Table 2 (continued)

Common	Variable	Description
A15	MD1	Successive preplanned movement index-1
A15	MD2	Successive preplanned movement index-2
	MINUTS	Sortie time
common	MLDARAS(I)	Number of loading docks for mission I
DAYTIM	MNTHST	Scheduled maintenance histories - Ø=No, 1=Yes
LIMITS	MNTIME	Minimum time increment allowed (1 minute)
DAYTIM	MOVED	Used in conjunction with transfer of elements of larger arrays
common	MOVRATE(I)	Movement rate for unit group I
DAYTIM	MPUNCH	Option to punch delayed mission simulation results (=1)
DAYTIM	MRQDAY	Current mission request day
DAYTIM	MRQDY1	Number of days of mission requests that have been generated
DAYTIM	MRQEST	Type of mission request now being serviced
DAYTIM	MRQTME	Current mission request time
DAYTIM	MRQTYP	Type of mission requested
DAYTIM	MRQUST	Unit move request index
common	MSLD(I)	Total loading time for priority I missions
DAYTIM	MSNAME(I)	Name of mission I
REQRYS	MSNDAY	Mission day
	MSNDEL	Missions delayed to date
	MSNDIST	Mission distance
DAYTIM	MSNDRP	Number of missions dropped today to prevent queue overflow
DAYTIM	MSNDRPT	Number of missions dropped to date to prevent queue overflow
common	MSNDSKM	Mission distance in kilometers
	MSNGRP	Mission group
common	MSNGRP(I)	Group handling mission I
common	MSNLOAD(I)	Loading rate (tons/minute) for mission I
	MSNPR	Mission priority
common	MSNPYLD(I)	Payload (tons) for mission I
A13	MSNRAD	Mission radius

Table 2 (continued)

Common	Variable	Description
LIMITS	MSNREQ	Maximum number of mission requests per time increment (=36)
common	MSNSDL(I,J)	Daily missions of type I priority J delayed
LIMITS	MSNTAG	Vehicle unavailable (on a mission) indicator
	MSNTYP	Mission type
common	MSNUNLD(I)	Unloading rate (tons/minute) for mission I
common	MSULD(I)	Total unloading time for priority I missions
common	MULARAS(I)	Number of unloading docks for mission I
DAYTIM	MVDPOOL	Pool move indicator
All	MVEMNT	Unit move indicator
POOLSP	MVPLSP	Pool supply point move indicator
LIMITS	MXDREQ	Maximum scheduled missions allowed
DAYTIM	MXFBDY	Maximum number of days in advance cycle
LIMITS	MXINCR	Maximum number of time increments per day (=144Ø)
LIMITS	MXMSNS	Maximum number of mission types allowed
LIMITS	MSNVEH	Maximum number of vehicles per type allowed
LIMITS	MXQUE	Maximum number of mission requests allowed in queue
LIMITS	MXSMS	Maximum number of steps for scheduled maintenance
LIMITS	MXUSMS	Maximum number of steps for unscheduled maintenance
LIMITS	MXVEHS	Maximum number of vehicle types
LIMITS	MXVHGP	Maximum number of sorties per mission
D7	MXVHPR	Maximum number of vehicle preferences
ACØ3	NACT	Total fleet size
VEHRY1	NACTYP	Temporary storage for number of vehicles of a type
	NAME1	Data name for heading titles
JOBTMS	NAMMOL	Number of ammunition links
VEHRY1	NAVTM	Vehicle availability time prior to assignment in a group
common	NAVTME(I,J)	Next time vehicle I of type J is available
JOBTMS	NCARGL	Number of cargo links
D7	NCDPRI	Encoded priority

Table 2 (continued)

Common	Variable	Description
LIMITS	NCDREQ	Mission type encode (=1ØØØ)
INTIME	NDAY	Number of integer hours in working day
REQRYS	NDETRQ	Number of scheduled mission requests this run
DAYTIM	NDMHSI	Number of frequency intervals for delayed missions
DAYTIM	NDPVA	Number of days per vehicle assigned
VHPØ1	NDVHS	Mission interval for printing of vehicle preference table
	NERR	Number of errors
	NEXTIM	Clock setting for next available request
VHPØ1	NFM9ØØ2	Format for printing vehicle preference data
DAYTIM	NMOVED	Number of units moved
	NONE	Homogeneous discipline indicator
	NONHOM	Error of homogeneity indicator
DAYTIM	NPLACE	Insertion place in queue for next scheduled or preplanned mission
DAYTIM	NPLSRC	Used in conjunction with transfer of elements of larger arrays
	NPO	Vehicle preference index
REQVAR	NPRIOR	Priority of mission request now being serviced
	NPX	Previous X coordinate of supply point
	NREQS	(Number of requirements-1) this time increment
common	NSCHMS(I,J)	Counters for scheduled and preplanned missions
DAYTIM	NSRCGPS	Number of SRC groups
DAYTIM	NSRCSGP	Number of SRCS per group
VEHRY1	NTHSTME(I)	Soonest available vehicle this day of type I
VEHRY1	NTIME(I)	New time beyond last day for vehicles of type I
VEHRY1	NTMTAG	Encode for vehicle time (=1,ØØØ,ØØØ)
	NT1	Print unit indicator
DAYTIM	NT2	I/O unit 2 designator
DAYTIM	NT3	I/O unit 3 designator
	NUMP	(=NUMPR)
DAYTIM	NUMPRS	Number of mission priorities this run
VEHRY1	NUMVEH(I)	Number of type I vehicles available this run

Table 2 (continued)

Common	Variable	Description
DAYTIM	NUNITS	Number of available SRC units
	NV	Vehicle preference index
VEHRY1	NVEH	Vehicle number of type now being used to fulfill request
VEHRY1	NVEHSQ(I)	Cumulative daily sum of squares of number of type I vehicles used
	NVH	Number of vehicles of all types used per day
VEHRY1	NVHDAY(I)	Daily number of type I vehicles used
VHPØ1	NVHPRFØ	Vehicle preference number
VEHRY1	NVHTOT(I)	Cumulative daily sum of number of type I vehicles used
VEHRY1	NVHTYP	Vehicle type now being used to fulfill request
	NVT	Number of vehicle types
RNPRIM	NXRNMK	Number of random numbers to discard from RAND2
RNPRIM	NXRNMQ	Number of random numbers to discard from RANDØ
RNPRIM	NXRNMS	Number of random numbers to discard from RAND1
RNPRIM	NXRNTM	Number of random numbers to discard from RAND3
RNPRIM	NXRNUM	Number of random numbers to discard from RANDØA
VEHRY1	NXTIME(I)	Soonest available vehicle of type I on next day
REQVAR	NXTVAL	Next available location in queue
VEHRY1	NXTVEH(I)	Next available type I vehicle
DAYTIM	N13	Number of mission types this run
DAYTIM	N42	Number of scheduled maintenance steps this run
DAYTIM	N54	Number of mission groups
DAYTIM	N95	Print option (=1 or 2) for vehicle operating and maintenance characteristics
DAYTIM	N98	End of job indicator - Ø = Not end of job, 1 = End of job
OUTSUM	OPERTM	Total operating time for all vehicles
DAYTIM	OTHER	Index to priority
A1Ø	OUT1(1,IREC) (2,IREC) (3,IREC)	=DEL =MRQEST =NPRIOR
common	OWNER(I,J) (1,J) (2,J) (3,J) (4,J)	Queue pointers =Location of first priority J request =Location of last priority J request =Location of current priority J request =Number of requests left for priority J

Table 2 (continued)

Common	Variable	Description
	PCTGPFT	Percentage of group-forming wait time this day
	PCTIDLE	Percentage of idle time this day
	PCTMNT	Percentage of maintenance time this day
	PCTOPER	Percentage of operating time this day
	PCTTATM	Percentage of turnaround time this day
VEHRY1	PM(I)	Probability of maintenance for a vehicle of type I
DAYTIM	POOLAG	Pool lag distance
DAYTIM	POOLEAD	Pool lead distance
DAYTIM	POOLXP	X-coordinate of pool
DAYTIM	POOLYP	Y-coordinate of pool
DAYTIM	PREVUSM	Previous largest unscheduled maintenance time for a vehicle
REQRYS	PRIOR	Priority of unscheduled mission request being generated, or priority of mission request being placed in the queue
DAYTIM	PRIORITY(I)	Priority of unit type I
All	PTOLDST	Distance from pool to old location
common	QUEUE(I,J)	Mission request data of priority J
	(1,J)	=Successor of this mission request
	(2,J)	=Type of this mission request
	(3,J)	=Time of this mission request
	(4,J)	=Day of this mission request
	(5,J)	=Predecessor of this mission request
DAYTIM	RDNWF	Road network factor
REQRYS	REQTI	Number of time increments in a mission request day
	REQTME	Request time in minutes
DAYTIM	RMSNDY	Scheduled and unscheduled missions per day counter
	RN	Random number
DAYTIM	RTTM	Return travel time
	RUNDLY	Total periods of delayed sorties
	RUTNAP	Variance of number of vehicles used per day of a type
common	SCHCO(I,J)	Number of times vehicle I of type J has entered scheduled maintenance

Table 2 (continued)

Common	Variable	Description
common	SCHIST(I,J)	Number of steps of scheduled maintenance vehicle I of type J has completed
OUTSUM	SCHMN	Scheduled maintenance time for all vehicles
VEHRY1	SCHT	Scheduled maintenance time per sortie
common	SCHTRS(I)	Scheduled maintenance threshold for type I vehicles
OUTSUM	SOPHRS	Total operating time for all vehicles
common	SOR(I,J)	Cumulative probability that a type I priority J unscheduled mission occurs
	SORGUP	Number of missions delayed
DAYTIM	SORTIM	Operating (travel) time for the sortie
common	SORTNM(I,J)	Total missions of type I priority J delayed
common	SRCLDM	Last day unit moved
common	SRCMPXD	Number of days during which unit must move once
common	SRCTYPE	SRC number
common	SRCXP	Original and current X-coordinate of unit
common	SRCYP	Original and current Y-coordinate of unit
ACØ3	SRVHRS	Service hours per day
DAYTIM	SRVTI	Service time increments per day
DAYTIM	SRVTME	Service minutes per day
DAYTIM	STATUS	Mission status indicator Ø = Mission is only partially filled 1 = Mission has been dispatched
REQRYS	SUMSOR(I)	Sum of square of daily missions of priority I completed
VEHRY1	TATM	Temporary storage for vehicle/mission turnaround time
common	TBTM(I,J)	Mean time between maintenance for vehicle type J on mission type I
	TEMP	Temporary storage of next available queue location
DAYTIM	THPER	Length of time increment in minutes
DAYTIM	THPERE	$=\text{THPER}-1^5$
DAYTIM	THPERH	Half length of time increment in minutes
REQRYS	TIMEE	Event time for unscheduled mission requests being generated

Table 2 (continued)

Common	Variable	Description
DAYTIM	TITLE(I)	7Ø-character page heading for simulation results
OUTSUM	TMAVNU	Total idle time for all vehicles
	TMSC	Total mission completion time (hours)
common	TMSCTM	Total mission completion time
OUTSUM	TNTUSD	Total idle time per vehicle type
common	TOPHRS(I,J)	Total operating time (minutes) for vehicle I of type J
common	TOSCH(I,J)	Time until a vehicle of type J goes into step I of scheduled maintenance
	TOT	Cumulative count of delayed missions
DAYTIM	TOTAL	Total number of sorties per day
common	TOTDST	Sum of mission distances for all units
OUTSUM	TOTLTM	Total available time for all vehicles
	TOTSNM	Total number of sorties delayed
OUTSUM	TOTTIM	Total time available per vehicle type
	TOTTTM	Total travel time across links
JOBTMS	TRAVTM(I,J)	Travel times by SRC unit by vehicle type
	TTM	Link travel times
JOBTMS	TTMCAL	Travel time - common ammunition links
JOBTMS	TTMCCL	Travel time - common cargo links
OUTSUM	TURNTM	Total turnaround time for all vehicles
common	TVHLD(I)	Total vehicle type I loading time
common	TVHSM(I)	Total vehicle type I scheduled maintenance
common	TVHTAT(I)	Total vehicle type I turnaround time
common	TVHUNL(I)	Total vehicle type I unloading time
common	TVHUSM(I)	Total vehicle type I unscheduled maintenance
VMWØ1	TVP	Used in conjunction with transfer of elements of larger arrays
DAYTIM	TYPES(I)	SRC for group entry I
LIMITS	TYPTAG	Vehicle type encode for group now forming
	UNAVTI	Time unavailable (hours)
DAYTIM	UNITMVE	Index to priority
DAYTIM	UNLDTM	Unloading time

Table 2 (continued)

Common	Variable	Description
VEHRY1	UST	Unscheduled maintenance time per sortie
common	VEHMSNS(I,J)	Number of type I missions done by vehicle type J
common	VEHPRF(I,J)	Preference of vehicle J doing request (mission) I
OUTSUM	VEHSAV	Total vehicle available during all days
OUTSUM	VEHSQ	Sum of squares of vehicles used daily
OUTSUM	VEHSUS	Total vehicles used for all days
common	VEHTAT(I)	Turnaround time for vehicle type I
OUTSUM	VEHUSD	Total vehicles per type used for all days
common	VHCOST(I)	Vehicle operating cost for vehicle type I
	VHLD	Total load time (hours)
common	VHPYLD(I)	Vehicle payload (tons) for vehicle type I
	VHSM	Total scheduled maintenance time (hours)
common	VHSPKPM(J)	Vehicle speed (KPH) for vehicle type I
	VHTAT	Total administrative time (hours)
	VHUNL	Total unloading time (hours)
	VHUSM	Total unscheduled maintenance time (hours)
common	VHWAIT(I)	Total group-forming wait time for vehicle type I
common	VOLF(I)	Vehicle overload factor for vehicle type I
	VSP	Vehicle speed
	WAIT	Time a vehicle has been waiting for the group to form
OUTSUM	WALTM	Group-forming time per vehicle type
DAYTIM	XCENTER	Used in conjunction with transfer of elements of larger arrays
DAYTIM	XLIMIT	Number of days for current run (=LIMIT)
DAYTIM	XLM1	Day of simulation-1
DAYTIM	YCENTR	Used in conjunction with transfer of elements of larger arrays

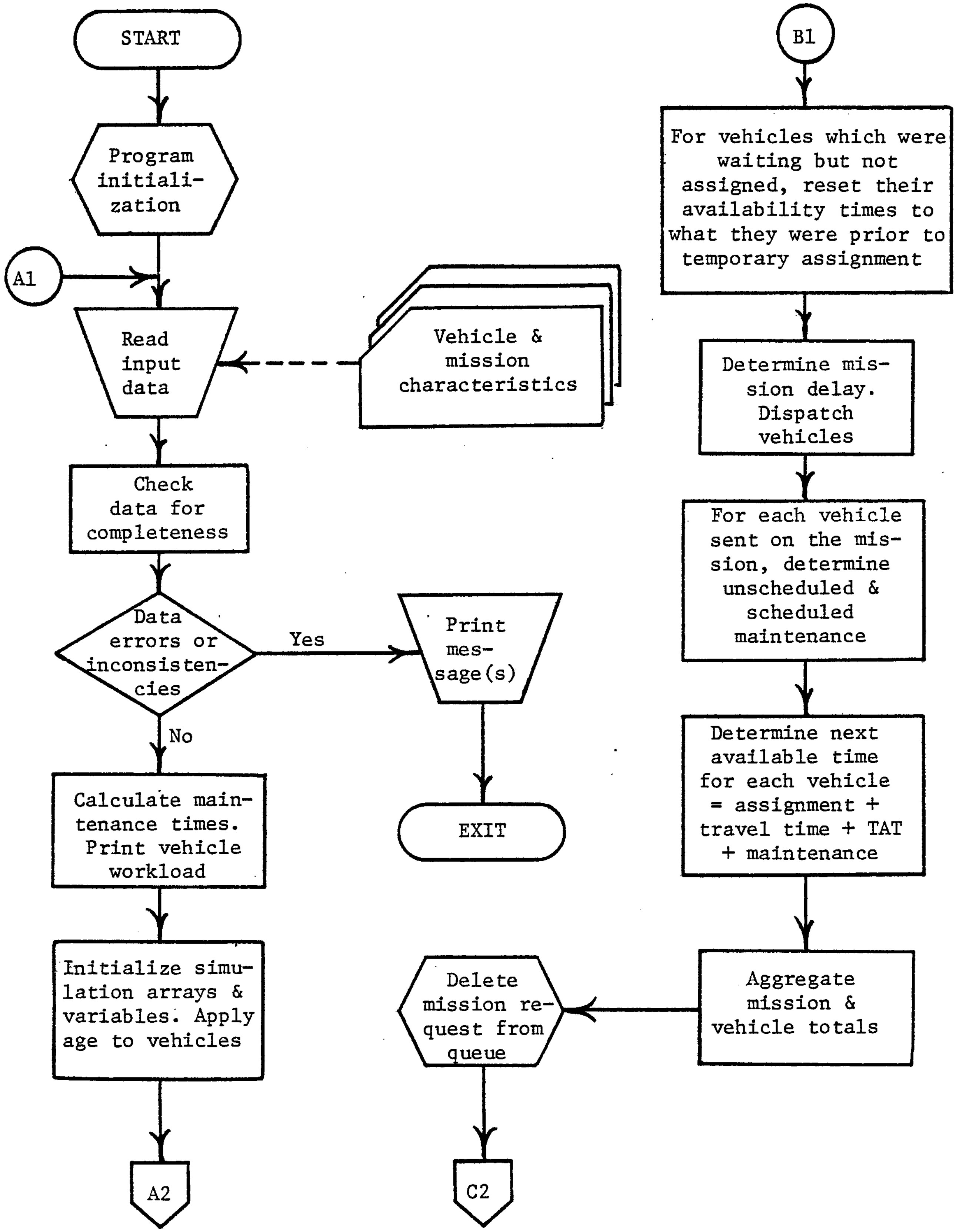

Figure 3—System Flow Diagram

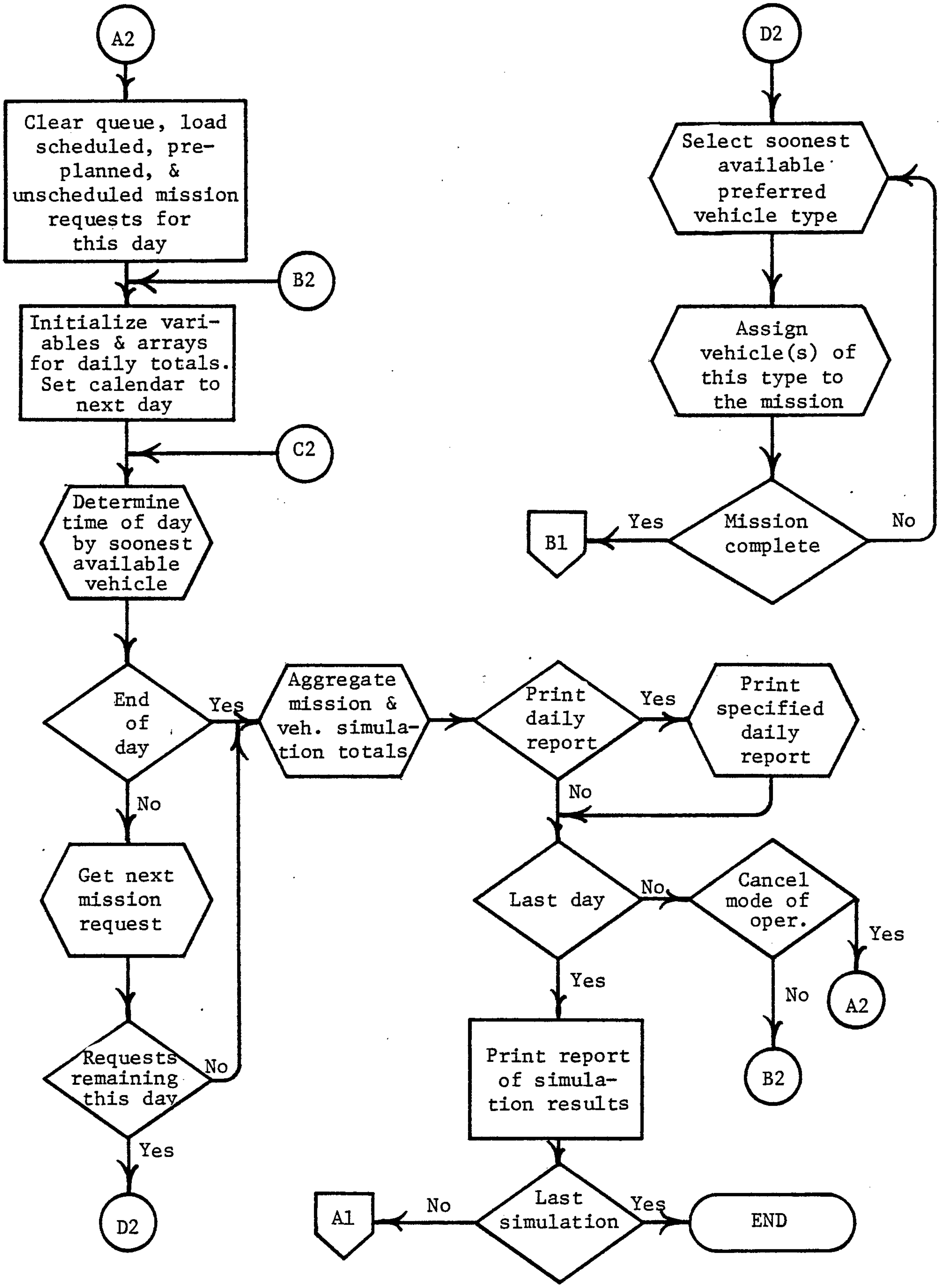

Figure 3 (continued)

mission. This process continues until mission vehicle requirements are satisfied. The availability times for vehicles which were waiting but not assigned are reset to what they were prior to temporary assignment. Mission delay time is then established, and the vehicles are dispatched to perform their missions. Scheduled and unscheduled maintenance is then determined for each vehicle sent on the mission, as well as the next available time for each vehicle. Mission and vehicle totals are aggregated, and the mission request is deleted from the queue. Processing of the next mission request, if any, continues beginning at the point where the time of day is determined by the soonest available vehicle, as previously noted.

If it is the end of the day or the request queue is found to be empty, then mission and vehicle simulation totals are aggregated and, depending upon a print option code, a daily report is printed. Until the last day of simulation is reached, another day is simulated beginning with clearing and loading of the mission request queue. When the last day of simulation is reached, simulation results are printed. Depending upon whether or not all simulations have been processed, the program either terminates or begins anew with another set of vehicle and mission characteristics as input.

MAIN ROUTINE AND CALLING SEQUENCE

This section indicates the calling sequence of model subroutines; that is, the order in which subroutines are called by the main program DSPTCHR and/or by subroutines. It should be noted that the calling of some subroutines is conditional, depending on the content of the problem data or on the options specified by the user. A brief description of the function of each subroutine is provided as an aid in following the general events which take place during the simulation.

1 SETUP1 - Initializes program and simulation variables. No subroutines are invoked by this subroutine.

2 CARDS - Controls editing and storing of all parameter cards. Edits and stores information from parameter cards. The following subroutines may be invoked by this subroutine:

2.1 CARDXX - Coverts the numeric quantities on all cards which are input to the simulation. No subroutines are invoked by this subroutine.

2.2 STORØ2 - Edits and stores the contents of a type 02 card. No subroutines are invoked by this subroutine.

2.3 STORØ4 - Stores contents of a type 04 card. No subroutines are invoked by this subroutine.

2.4 STOR11 - Stores contents of type 11 cards. The following subroutines may be invoked by this subroutine:

2.4.1 TMINCR - Converts clock time (0001-2400) to time increments (1-1440). No subroutines are invoked by this subroutine.

2.4.2 CARDXX - See 2.1.

2.5 STOR12 - Stores the contents of a type 12 card. The following subroutines may be invoked by this subroutine:

2.5.1 SRTPER - Generates cumulative probability distribution (Poisson) for each priority of unscheduled mission requests. No subroutines are invoked by this subroutine.

2.6 STOR13 - Forms a cumulative probability distribution for each type of unscheduled mission request within priority. The following subroutine may be invoked by this subroutine:

2.6.1 ST1325 - Edits and stores the contents of type 12 and type 25 cards. The following subroutine may be invoked by this subroutine:

2.6.1.1 CARDXX - See 2.1.

2.7 ISTOR2 - Accepts and validates data of card types 15 and 55. No subroutines are invoked by this subroutine.

2.8 FSTOR1 - Validates input card types 18, 22, 31, 32, 41, 51, 52, 53, 73, 76, 77, and 82. No subroutines are invoked by this subroutine.

2.9 RDPOOL - Reads pool map coordinates. The following subroutine is invoked by this subroutine:

2.9.1 CNVXY - Converts the lead and lag distances of the vehicle pool to kilometers. No subroutines are invoked by this subroutine.

2.10 ISTOR1 - Accepts and validates input data of card types 21, 54, 72, 81, and 83. No subroutines are invoked by this subroutine.

2.11 STOR25 - Forms a cumulative probability distribution for each type of unscheduled mission request within priority. The following subroutines are invoked by this subroutine:

2.11.1 ST1325 - See 2.6.1.

2.11.2 SRTPER - See 2.5.1.

2.12 FSTOR2 - Validates input card types 42 and 44. No subroutines are invoked by this subroutine.

2.13 STOR43 - Edits and stores the contents of type 43 cards. The following subroutine may be invoked by this subroutine:

2.13.1 CARDXX - See 2.1.

2.14 RDSRCS - Reads and stores the coordinates of units at the start of simulation. The following subroutines may be invoked by this subroutine:

2.14.1 CARDXX - See 2.1.

2.14.2 FSTOR1 - See 2.8.

2.14.3 CNVXY - See 2.9.1.

2.14.4 TYPRATE - Determines move frequency for unit (IU) based on unit type. No subroutines are invoked by this subroutine.

2.15 CALCDST - Calculates distance of organizational units from pool for initial placement. No subroutines are invoked by this subroutine.

2.16 RDTTM - Reads travel time data for each vehicle type over common and unique links between supply points and unit locations. No subroutines are invoked by this subroutine.

2.17 PRNTIN - Prints the page heading, length of simulation, and operation modes for the simulation. The following subroutine may be invoked by this subroutine:

2.17.1 PRNTN2 - Prints table of sorties per mission and vehicle preference tableau. The following subroutine may be invoked by this subroutine:

2.17.1.1 MSPRCT - Adjusts the count of maximum mission priorities and maximum mission types for the current run. No subroutines are invoked by this subroutine.

2.18 PRNTN1 - Prints maintenance history for vehicle types and number of vehicles available. No subroutines are invoked by this subroutine.

3 CTEDPPM - Edits and counts preplanned mission requests. The following subroutines may be invoked by this subroutine:

3.1 CARDXX - See 2.1.

3.2 PRNTIN - See 2.17.

3.3 PRNTN1 - See 2.18.

3.4 PRNTQUE - Prints the contents of the input or mission request queue by priority. No subroutines are invoked by this subroutine.

4 MSPRCT - See 2.17.1.1.

5 VHMSPRF - Assigns vehicles to missions based on mission priority and vehicle preference. The following subroutine may be invoked by this subroutine.

5.1 SRTPRF - Sorts vehicles available to perform in a mission by preference, if a preference is specified. No subroutines are invoked by this subroutine.

5.2 STVHPF - Designed to determine and store the preferences for vehicle involvement in mission requests. No subroutines are invoked by this subroutine.

6 MINVEHS - Ensures mission definition (number of sorties, travel time, turnaround time, preference) for each permissible vehicle type. The following subroutines may be invoked by this subroutine.

6.1 MSNPCT - Calculates average maintenance time per operating hour and average time between maintenance for a particular vehicle type. No subroutines are invoked by this subroutine.

6.2 MNTVEH - Computes vehicle travel times and maintenance times by vehicle type. No subroutines are invoked by this subroutine.

7 CNTRLØ - Sets the random number generators and initializes array variables. No subroutines are invoked by this subroutine.

8 CNTRL2 - Initializes mission group forming areas. The following subroutines may be invoked by this subroutine:

8.1 CLRQUE - Clears the mission request queue and associated variables. No subroutines are invoked by this subroutine.

8.2 DAYREQ - Controls the loading of one day of mission requests into the mission queue. The following subroutines may be invoked by this subroutine:

8.2.1 NEXTIN - Loads the type, time, and day of a mission request into the mission request queue. The following subroutine may be invoked by this subroutine:

8.2.1.1 DELAST - If applicable, deletes, from the queue, the first mission request of the lowest priority. The following subroutine may be invoked by this subroutine:

8.2.1.1.1 DELETE - Deletes, from the mission request queue, the request of a particular location and priority. No subroutines are invoked by this subroutine.

8.2.2 SCHMSN - Controls retrieval of daily scheduled mission requests. The following subroutines may be invoked by this subroutine:

8.2.2.1 PRNTQUE - See 3.4.

8.2.2.2 QUESCH - Uses the time and day of the next scheduled or preplanned mission request, chronologically within priority, to find an insertion point. No subroutines are invoked by this subroutine.

8.2.2.3 INSERT - Inserts a specified mission request into the mission request queue according to priority and time sequence. The following subroutine may be invoked by this subroutine:

8.2.2.3.1 DELAST - See 8.2.1.1.

8.2.3 PPMSNS - Controls daily entry of preplanned missions into the mission request queue. The following subroutines may be invoked by this subroutine:

8.2.3.1 PRNTQUE - See 3.4.

8.2.3.2 STORPP - Retrieves preplanned mission requests and controls placement of these requests into the mission request queue. The following subroutines may be invoked by this subroutine:

8.2.3.2.1 CARDXX - See 2.1.

8.2.3.2.2 TMINCR - See 2.4.1.

8.2.3.2.3 QUESCH - See 8.2.2.2.

8.2.3.2.4 INSERT - See 8.2.2.3.

8.2.4 MOVPOOL - Examines mission requests to see if the pool has moved before the start of the next day. The following subroutine may be invoked by this subroutine:

8.2.4.1 CALCDST - See 2.15.

8.2.5 MOVSRCS - Determines eligibility of unit to be transferred to the mission request queue. The following subroutines may be invoked by this subroutine:

8.2.5.1 QUESCH - See 8.2.2.2.

8.2.5.2 INSERT - See 8.2.2.3.

8.2.6 PRNTQUE - See 3.4.

9 CNTRL4 - Sets vehicle assignment clock and clears counters of daily missions. No subroutines are invoked by this subroutine.

10 CNTRL3 - Maintains the time, initiates mission requests, and ends a simulated day. The following subroutine may be invoked by this subroutine:

10.1 NEXTME - Finds the soonest time a vehicle of any type will be available. No subroutines are invoked by this subroutine.

10.2 NXTREQ - Selects the next mission request to be filled. The following subroutines may be invoked by this subroutine:

10.2.1 NXAVPAC - Finds the soonest available vehicle type which can perform the mission currently requested. No subroutines are invoked by this subroutine.

10.2.2 DAYREQ - See 8.2.

10.2.3 DAYOUT - Provides daily reports for all vehicle types. The following subroutines may be invoked by this subroutine:

10.2.3.1 ADJTME - Calculates the time used by vehicles beyond the last day of simulation. No subroutines are invoked by this subroutine.

10.2.3.2 DAYOU3 - Keeps a tally on the number of each type of vehicle available to the simulation and of all operating and down times. No subroutines are invoked by this subroutine.

10.2.4 OUTPT1 - Aggregates vehicle waiting time and vehicle operating time. The following subroutine may be invoked by this subroutine.

10.2.4.1 VEHSUM - Accumulates grand totals for all vehicle operating (travel) time, maintenance time, group-forming time, and idle time. No subroutines are invoked by this subroutine.

10.2.5 OUTPT2 - Generates statistics on mission activity. No subroutines are invoked by this subroutine.

10.2.6 OUTPT3 - Generates print data for mission vehicle group-forming time per mission. No subroutines are invoked by this subroutine.

10.2.7 OUTPT4 - Prints data concerning mission completions and delays. No subroutines are invoked by this subroutine.

10.2.8 MOVSRC - Performs the function of moving an SRC (unit) when time to move has elapsed. No subroutines are invoked by this subroutine.

10.2.9 DELETE - See 8.2.1.1.1.

10.3 ASSMNT - Finds the next available vehicle(s) within a vehicle type. The following subroutine may be invoked by this subroutine:

10.3.1 ASMNT3 - Assigns the next available vehicle, according to its type, to a mission group. The following subroutine may be invoked by this subroutine.

10.3.1.1 ASMNT1A - Recalculates the available time for a particular vehicle of a particular type. No subroutines are invoked by this subroutine.

10.3.1.2 ASMNT1 - Dispatches the soonest available group of vehicles to fulfill the current mission request. The following subroutines may be invoked by this subroutine:

10.3.1.2.1 MOVSRC - See 10.2.8.

10.3.1.2.2 ASMNT1A - See 10.3.1.1.

10.3.1.2.3 ASMNT1B - Calculates group-forming waiting time for the current mission request. The following subroutine may be invoked by this subroutine:

10.3.1.2.3.1 ASMNT2 - Aggregates travel time for a particular vehicle type. No subroutines are invoked by this subroutine.

10.3.1.2.4 WRMSDL - Fills an array with mission type, mission priority, and mission delay data. No subroutines are invoked by this subroutine.

10.3.1.3 DELETE - See 8.2.1.1.1.

10.4 NXAVPAC - See 10.2.1.

11 CNTRL5 - Resets vehicle clocks to the next day. The following subroutines may be invoked by this subroutine:

11.1 PRVHAV - Assigns each vehicle in the simulation a unique number, based on vehicle type. No subroutines are invoked by this subroutine.

12 DAYOUT - See 10.2.3.

13 POSTURE - Reads the next SRC deck, in order to check for a change in posture of the SRC. The following subroutine may be invoked by this subroutine:

13.1 NEWSRCS - Computes new coordinates of an SRC when a unit move has been made. The following subroutines may be invoked by this subroutine.

13.1.1 CARDXX - See 2.1.

13.1.2 FSTOR1 - See 2.8.

13.1.3 TYPRATE - See 2.14.4.

14 DAYREQ - See 8.2.

15 ADJTME - See 10.2.3.1.

16 CALCDST - See 2.15.

17 OUTPT1 - See 10.2.4.

18 OUTPT2 - See 10.2.5.

19 OUTPT3 - See 10.2.6

20 OUTPT4 - See 10.2.7.

21 RDMSDL - Reads the mission delay input data. No subroutines are invoked by this subroutine.

22 OUTPT5 - Prints frequency distributions of delayed missions and mission completion times. The following subroutine may be invoked by this subroutine:

22.1 MSNCTPR - Calculates delay time for percent of completed missions by priority. No subroutines are invoked by this subroutine.

23 OUTPT6 - Prints frequency distributions of delayed missions, and mission completion times. The following subroutine may be invoked by this subroutine.

23.1 MSNCTGP - Calculates delay time for percent of completed missions by mission group. No subroutines are invoked by this subroutine.

24 RDSRCS - See 2.14.

SUBROUTINE DESCRIPTIONS

The complete program package consists of a main control routine and a series of 74 subroutines, together with simulation functions. This section contains a brief narrative description of each subroutine; their flow charts and listings appear in appendices to this volume. All routines are coded in FORTRAN V for compilation under the CDC FTN compiler.

DSPTCHR

(Main Routine). Controls all major program activities such as: initialization of program variables, reading of vehicle-mission data, simulation of each day of time period, printing of daily results, printing of final results, and cycling from one simulation case to another.

ADJTME

Calculates the time used by vehicles beyond the last day of simulation. This usage results from vehicles which have been dispatched on a mission but have not yet returned when the simulation ends. The vehicles may still be on a mission or they may be in maintenance.

When dispatching a vehicle at 0900 hours, the following is done to determine the vehicle's next available time:

0900 ▽ T-A-T ▽ 0930 Travel ▽ 1130 T-A-T ▽ 1200 Travel ▽ 1400 Maintenance ▽ 1600

The times for which the vehicle is expended are aggregated and added to the present time to give the next available time for the vehicle. In this case, one hour turnaround time, four hours travel time, and two hours maintenance time when added to 0900 hours give the vehicle its next available time of 1600 hours. If the simulation ended at 1200 hours, the time to 1600 hours needs to be considered as part of the total time simulated in order to give an accurate account of vehicle utilization in the results.

ASMNT1

Dispatches the soonest available group of vehicles to fulfill the current mission request. Returns to the vehicle pool for reassignment any vehicles of other groups which may have been waiting. The availability times for these latter vehicles are reset to that time when they were first

selected in an attempt to fill this mission. This subroutine also maintains by priority and type aggregations of mission delay times and counts of missions that are completed and delayed.

<u>ASMNT1A(I,J)</u>

Recalculates the availability time for vehicle I of type J, which is returned to the pool because its group was not the soonest available and hence not dispatched.

<u>ASMNT1B(I,J)</u>

Calculates group-forming waiting time for the current mission request and determines maintenance times for vehicles dispatched on the mission. Determines the next available time for these vehicles and returns them to the pool. Aggregates vehicle workload and waiting times for vehicles of type J.

<u>ASMNT2(J)</u>

Aggregates travel time for vehicle type J. Determines unscheduled and scheduled maintenance times. Aggregates maintenance times and checks vehicle's scheduled maintenance threshold if it has undergone unscheduled maintenance.

<u>ASMNT3</u>

Assigns the next available vehicle, according to its type, to a mission group. Saves the time of assignment to this group, the vehicle number, and the vehicle's availability time at the time of assignment. Tags vehicle as belonging to a group currently forming. Adjusts the mission group size to include the newly assigned vehicle. When a mission group becomes complete, an indicator is set. Dispatches the mission. Deletes from the mission queue the mission request just filled.

<u>ASSMNT</u>

Finds the next available vehicle(s) within a vehicle type. If possible, dispatches the mission with vehicles of this type; otherwise, it returns to the controller (CNTRL3) to wait for another vehicle type from which it again selects the next available vehicle(s).

BLOCK DATA

Predefines program constants and limits of subscripted variables and arrays.

CALCDST

Calculates distance of organizational units from pool for initial placement. Also recomputes coordinates at pool when a move occurs. Grid location is printed if print option is selected. Not exercised for HIMO simulations.

CARDS

Controls editing and storing of all parameter cards. Edits and stores information from parameter card types 1, 3, 7, 65, 71, 95, 96, 97, and 98. Control is maintained by means of a card index. Normally, the index is the quantity in the first one or two columns of the card. The index is followed by a comma or blank which delimits the first data field of the card. For example:

```
15,01,02,4,5        96
```

The indexes on the above cards are 15 and 96, respectively.

CARDXX

Converts the numeric quantities on all cards which are input to the simulation. Quantities, which are also considered as data fields, are defined to be those strings of digits between: (1) commas, (2) column one and the first comma or blank, and (3) the last comma and first blank or end of the card. A blank or end of card terminates conversion for the card. A mispunch in any quantity or data field terminates its conversion and control skips to the next quantity. The mispunched quantity is set to zero.

Converted data are stored into successive locations, starting with the first location in the array INPUT. In successive locations of the array INTYPE, a 1 implies that the corresponding location in the INPUT array contains a floating point quantity; a zero implies an integer quantity.

CLRQUE

Clears the mission request queue and associated variables. Each mission request is described by five variables:

1. Successor (location of next request)
2. Mission request type
3. Mission time
4. Mission day
5. Predecessor (location of preceding request)

There are four descriptors set aside to govern each priority. They are:

1. Location of first mission request
2. Location of last mission request
3. Location of current mission request
4. Number of requests in this priority

CNTRLØ

Sets the random number generators. Initializes array variables which accumulate vehicle operating and maintenance times during the entire simulation period. Sets clocks of all vehicles to zero. Controls the aging of vehicles. Clears storage required for accumulating mission statistics and vehicle statistics. Clears vehicle workload storage. Clears mission load and unload times.

CNTRL1

Loads the scheduled maintenance history or age into each vehicle array. Several steps or levels of scheduled maintenance can be specified. A step or level first consists of a time duration for which the vehicle operates (travels) free of scheduled maintenance. A second time is the duration which the vehicle spends in maintenance after the first time duration has elapsed.

For example, if five levels of scheduled maintenance with times in hours are:

Maintenance Steps	Time between Maintenance Actions	Duration of Maintenance Action
1	40	0
2	30	10
3	20	10
4	10	20
5	100	40

then a vehicle which is given a scheduled maintenance history of 2 has finished two steps and is beginning the third. Consequently it operates (travels) for 20 hours after which time it is put into maintenance for 10 hours. Another vehicle which has a history of 0, operates for 70 hours (levels 1 and 2) and then goes into maintenance for 10 hours. An option of this type also prevents all vehicles from entering scheduled maintenance concurrently.

CNTRL2

Initializes mission group forming areas. Clears mission request queue (prevents mission request carryover implying cancel mode). Loads mission request queue with missions for the current day (every day in cancel mode; only first day for carryover mode). Removes tag from any vehicles which may have been waiting in a mission group from the previous day (implies mission cancelled).

CNTRL3

Controls the simulation day. Maintains the time; initiates mission request; services request by controlling the assignment of soonest available preferred vehicles; and ends a simulated day when either the mission requests are finished or all vehicles are in use until the next day.

CNTRL4

Sets vehicle assignment clock to beginning of the day. Clears counters of daily missions delayed and completed within mission priority and within mission type and priority. Initializes soonest time vehicles of each type will become available.

CNTRL5

Resets vehicle clocks to the next day. Vehicle clock time is kept in time decrements. Therefore, a 24-hour day with 5-minute time decrements would reduce each clock by 288 (24 x 60)/5. Any negative clock times which imply vehicle idle time are set to zero. This subroutine aggregates by type: vehicles used per day and square of vehicles used per day (required in computation of standard deviation). It also aggregates missions completed per day within priority and within priority and type. And, it aggregates square of missions completed per day (for standard deviation computation). Finally, it aggregates missions dropped from queue to prevent overflow.

CNVXY

Converts the lead and lag distances of the vehicle pool to kilometers.

CTEDPPM

Edits and counts preplanned mission requests. Computes average daily frequency within mission priority and type. Adds daily averages to the count of scheduled mission requests which are used in maintenance computations. Prints average daily frequency of preplanned missions by priority and type.

DAYOUT

Provides daily reports with two options. For all vehicle types, lists vehicle availability, operating time, and maintenance history. Lists by priority and type the number of missions completed and delayed and their corresponding delay time to date (option 1), and the same statistics for the current day only (options 1 and 2).

DAYOU3

Keeps a tally on the number of each type of vehicle available to the simulation, and of all operating and down times. Computes utilization rates of each vehicle type, and keeps track of the time a particular vehicle type is unavailable. Also retains the amount of idle time a vehicle may have.

DAYREQ

Controls the loading of one day of mission requests into the mission queue. Mission requests are sequenced by time within priority. For those requests which occur at the same time, scheduled missions are sequenced before preplanned missions and then unscheduled missions follow last. This routine generates unscheduled mission requests by a multiple Poisson process. It steps through each time increment of a day and determines the number of events (unscheduled mission requests) which occur during each time increment. The event frequency is determined by the inverse transformation method using a cumulative Poisson probability distribution.

For example, let the cumulative Poisson probability distribution of priority 3 in the sample problem (see Table 3) be used as the distribution. Then, if a random number is drawn, say 0.75, no events occur at this time since it is less than 0.820438, which is the probability of zero events. If the next random number is 0.999, then three events will occur at this time since, probability (two events) $< 0.999 <$ probability (three events).

When events occur, the type of each must be determined. Again, this involves the use of the inverse transformation method. If five mission types can occur within a priority and their frequencies are: 10, 5, 15, 15, and 5, then the following cumulative probability distribution is formed: 0.20, 0.30, 0.60, 0.90, and 1.00. If it is known that two events must occur and the random numbers drawn are 0.26 and 0.81, then a type 2 and a type 4 mission request are generated since: probability (type 1) $<$ $0.26 \leq$ probability (type 2), and probability (type 3) $< 0.81 \leq$ probability (type 4). When the parameter $i \neq 0$, mission requests are only generated and not loaded into the queue.

DELAST

Used only when the mission request queue is full. Deletes from the queue the first mission request in the lowest priority (implies the longest unserviced request of least importance). Its storage then becomes available for a new request. This procedure is necessary to prevent queue overflow which could occur when the vehicle fleet size is close to the minimum or during unusually high mission demand periods.

Table 3

CUMULATIVE POISSON PROBABILITY DISTRIBUTION[a]

K	P(K)	K	P(K)
0	.820438220141	12	.999999999966
1	.982816617877	13	.999999999969
2	.998885317486	14	.999999999972
3	.999945401044	15	.999999999975
4	.999997853293	16	.999999999979
5	.999999929528	17	.999999999982
6	.999999998015	18	.999999999985
7	.999999999951	19	.999999999988
8	.999999999954	20	.999999999991
9	.999999999957	21	.999999999994
10	.999999999960	22	.999999999997
11	.999999999963	23	1.000000000000

[a]Priority 3 unscheduled mission requests are generated using the Poisson distribution.

DELETE(I,J)

Deletes from the mission request queue that request in location I of priority J. The required storage for that request is then added to the list of available queue storage. It becomes the last location of available storage.

FSTOR1

Validates input card types 18, 22, 31, 32, 41, 51, 52, 53, 73, 75, 76, 77, and 82. For each MXTYPE number of cards of the above type, an array is filled. Error records are printed and severity of the error determines whether execution contines.

FSTOR2

Validates input card types 42 and 44. For each input card of this form, field positions are verified and stored in arrays. Error data are printed and severity of the error determines whether processing continues.

GRPSUM(I)

For vehicle types which may be combined to satisfy the current mission request, this routine aggregates the vehicles waiting within each type to determine if the group currently formed is sufficient to dispatch the mission. Each vehicle contributes to the group a fractional part which equals the reciprocal of its group size. For example, if a mission can use any combination of three vehicle types and it requires four, eight, and four of each type, respectively, then the mission could be dispatched when:

1. Eight type 2 vehicles are available (8/8 = 1), or

2. Two type 1 vehicles and two type 3 vehicles are available (2/4 + 2.4 = 1), or

3. One type 1 vehicle, four type 2 vehicles, and one type 3 vehicle are available (1/4 + 4/8 + 1/4 = 1), etc.

IGPSRV

Checks the validity of mission identifications to ensure that they are within the range of 1 to N13 (the maximum number of missions allowed). The vehicle type code is also checked to see that it is within the range of 1 to ITYPE (the maximum number allowed for an input vehicle number).

INSERT

Used for scheduled and preplanned mission requests. Inserts a specified mission request into the mission request queue according to priority and time sequence. If the request occurs at the same time as another mission request within its priority, it will precede that request. If the queue is full, the next request of the lowest priority will be dropped to make room for the new request.

ISTOR1

Accepts and validates input data of card types 21, 54, 72, 81, and 83. Data fields which are in error are flagged and displayed. The severity of the error determines whether processing continues.

ISTOR2

Accepts and validates data of card types 15 and 55. Data fields in error are flagged and printed. Processing continues based on the severity of the error.

MINVEHS

Ensures mission definition (number of sorties, travel time, turnaround time, preference) for each permissible vehicle type. Computes average sortie time for each vehicle type based on its mission workload (missions for which it is preferred). Calculates minimum number of vehicles of each type required to accomplish its mission workload. Controls printing of mission workload and maintenance times for each vehicle type.

MNTVEH(K)

For vehicle type K, calculates average sortie time, probability of maintenance, and expected time in maintenance given that maintenance occurs.

MOVPOOL

Examines mission requests to see if the pool has moved before the start of the next day. If movement has occurred, the routine prints both old and new coordinates of the pool. This ensures that the supply point does not move on the first day of simulation. If a preplanned move occurs, a print routine acknowledges the move. Not used for HIMO simulations.

MOVSRC

Performs the function of moving an SRC (unit) when time to move has elapsed. A print subroutine lists the unit number and the new grid coordinates, and notes the data and payload requirements of the move. Not used for HIMO simulations.

MOVSRCS

Determines eligibility of a unit to be transferred to the mission request queue. Inserts mission request into queue according to payload, priority, and time sequence. Accumulates number of missions requested per day by priority of unscheduled mission request. Not used for HIMO simulations.

MSNCTGP

Calculates delay time for percent of completed missions by mission group.

MSNCTPR

Calculates delay time for percent of completed missions by mission priority.

MSNPCT(K)

Calculates average maintenance time per operating hour and average time between maintenance for vehicle type K. Computes number of missions and sorties required of vehicle type K. Determines what percentage of its workload each mission represents. Prints mission workload tableau for vehicle type K (see MINVEHS).

MSPRCT

Adjusts the count of maximum mission priorities and maximum mission types for the current run to include any scheduled or preplanned mission priorities or mission types.

NEWSRCS

Computes new coordinates of an SRC when a unit move has been made. Processes input data for SRC move rates per day. Verifies that all SRC data are consistent and properly terminated. Prints a table, by SRC type, showing X and Y grid coordinates of unit since its last move. Not used for HIMO simulations.

NEXTIN

Loads the type, time and day of a mission request into the mission request queue. The mission is put last in its priority list. The method used to generate unscheduled missions produces the missions in chronological order; therefore, they only need to be inserted at the end of their respective priority group.

NEXTME

Finds the soonest time a vehicle of any type will be available. The list of soonest available times for each vehicle type is compared and the minimum time that is found determines the soonest available vehicle type.

NXAVPAC

Finds the soonest available vehicle type which can perform the mission currently requested. The list of availability times for only the preferred (capable and permissible) vehicle types is searched and the minimum time found, thus determining the soonest available preferred vehicle type.

NXTREQ

Selects the next mission request to be filled. Selection is determined on a first-come, first-served basis according to mission request priority and vehicle preference.

OUTPT1

Aggregates vehicle waiting time and vehicle operating time. Determines number of vehicles utilized. Aggregates vehicle turnaround time, scheduled and unscheduled maintenance time, and idle time. Calculates average number of vehicles utilized and their standard deviation. Prints report showing these computations.

OUTPT2

Generates statistics on mission activity, to include periods of delayed sorties and average delay time for each priority mission, as well as average number of priority missions completed each day and missions not completed. Standard deviations of all delays, completed and not completed missions, and demand rates are generated and printed.

OUTPT3

Generates print data for mission vehicle group forming time per mission, as well as average delay time per mission. Converts total mission load and unload times to average load/unload times. Prints message acknowledging that a mission was dropped during the run, if any were dropped.

OUTPT4

Produces output data dealing with average delay times and standard deviations of load/unload times. Prints data concerning mission completions and delays.

OUTPT5

Converts mission completion time to hours. Prints the frequency distributions of delayed missions, mission completion times, and amount of delayed tonnage.

OUTPT6

Produces abbreviated daily mission completion tableau. Based on print options, prints summary of frequency distributions for delayed mission and delayed tonnage.

POSTURE

Performs a read of the next SRC deck in order to check for a change in posture of the SRC.

PPMSNS

Controls daily entry of preplanned missions into mission request queue.

PRNTIN

Prints the page heading, length of simulation, and operation modes for the simulation. Prints the time ranges for servicing requests and assigning vehicles. Prints unscheduled mission probability distributions and daily mission frequencies. Prints time and frequency for scheduled missions.

PRNTN1

Prints the steps of scheduled maintenance for each vehicle type. Prints the scheduled maintenance history or number of steps of scheduled maintenance completed for each vehicle. Prints number of vehicles available within each type.

PRNTN2

Prints table of sorties per mission and vehicle preference tableau. Prints table of vehicle travel (operating) time and turnaround time per mission. Prints table of unscheduled maintenance thresholds per vehicle type.

PRNTQUE

Prints the contents of the input or mission request queue by priority. The contents are listed line-by-line, each line containing a location number, mission request successor, mission type, mission time, mission day, mission request predecessor, and location number.

PRVHAV

Assigns each vehicle in the simulation a unique number based on vehicle type.

QUESCH

Uses the time and day of the next scheduled or preplanned mission request chronologically within priority to find an insertion point. If a mission request occurs at the same time as one already in the queue, the new request will precede the one already in the queue (last-in, first-out when request times are coincident).

RANDO(M)

Generates uniform random variates between 0 and 1. When unscheduled mission requests are formed, this generator is used in determining the number of missions occurring during a time increment. Initially $M \neq 0$, so that the seed of the generator can be set and a prespecified quantity (see parameter card type 03) of numbers can be discarded before the generator is used. When $M = 0$, the next random number in the sequence is generated.

RANDOA(M)

Generates uniform random variates between 0 and 1. This generator is used in determining the unscheduled maintenance time per sortie. Initially $M \neq 0$, so that the seed of the generator can be set and a prespecified quantity (see card type 03) of numbers can be discarded before the generator is used. When $M = 0$, the next random number in the sequence is generated.

RAND1(M)

Generates uniform random variates between 0 and 1. This generator is used to determine when unit movements occur. Initially $M \neq 0$, so that the seed of the generator can be set and a prespecified quantity (see parameter card type 03) of numbers can be discarded before the generator is used. When $M = 0$, the next random number in the sequence is generated.

RAND2(M)

Generates uniform random variates between 0 and 100. When unscheduled mission requests are formed, this generator is used in determining the type for each unscheduled mission request. Initially $M \neq 0$, so that the seed of the generator can be set and a prepsecified quantity (see parameter card type 03) of numbers can be discarded before the generator is used. When $M = 0$, the next random number in the sequence is generated.

RAND3(M)

Generates uniform random variates between 0 and 1. The number generated is used to determine the number of time increments to elapse between resupply request times. Initially $M \neq 0$, so that the seed of the generator can be set and a prespecified quantity (see parameter card type 03) of numbers can be discarded before the generator is used. When $M = 0$, the next random number in the sequence is generated.

RMDSDL

Reads the mission delay input data and records the mission type, mission priority, and the frequency interval for delayed missions.

RDPOOL

Reads pool map coordinates, converts them to maintain an updated pool map and, dependent on the print option, prints coordinate values both before and after conversion. Not used for HIMO simulations.

RDSRCS

Reads SRC data and validates the values to ensure that priorities are assigned correctly and that payloads are within permissible range of values. The routine also guarantees that units which had been previously entered in the simulation are present, and that a record is maintained of the last day in which a unit moved. Exception messages are generated when inconsistencies occur in the data and when data are out-of-range.

RDTTM

Reads travel time data for each vehicle type over common and unique links between supply points and unit locations. Travel times are specified by SRC unit by vehicle and distances corresponding to each supply point-SRC unit. These data may vary according to scenario, snapshot, route definition, and weather type for all combinations of common and unique links.

SCHMSN

Controls retrieval of daily scheduled mission requests and placement of these requests into the mission request queue according to priority and time.

SETUP1

Initializes program and simulation variables. Presets simulation parameters that usually do not vary between simulation runs and that require definition even if not normally used. The routine prints a list of current program maximums or limits, initializes vehicle costs, load limits, turnaround time, and vehicle maintenance parameters.

SRTPER

Generates cumulative Poisson probability distribution for each priority of unscheduled mission requests. The probability of x occurrences is given by the Poisson distribution:

$$f(x) = \frac{e^{-\lambda} \lambda^{x}}{x!}$$

where

λ = unscheduled mission frequency per time increment.

The cumulative probability is given by:

$$F(x) = \sum_{k=0}^{x} \frac{e^{-\lambda} \lambda^{k}}{k!}$$

The cumulative distribution is developed in the program by the following relationship:

$F(x) = f(x) + F(x-1)$, for $x \geq 1$

$F(x) = f(x)$, for $x = 0$

<u>SRTPRF</u>

Sorts vehicles available to perform in a mission by preference, if a preference is specified.

<u>STORPP</u>

Retrieves preplanned mission requests for each day and controls placement of these into the mission request queue according to priority and time. The preplanned mission requests are used as a circular list in that the first day is reused after the last day of requests. If there are n days of preplanned mission requests, then the missions requested on the first day are also requested on day 1 + n, 1 + 2n, 1 + 3n, ..., etc. In this way if n = 7, identical weeks of preplanned activity can be simulated.

<u>STORØ2</u>

Edits and stores the contents of a type 02 card. Converts mission request times, mission dispatch times, and times of the working day from clock time (0001-2400) to minutes and time increments. Computes program time constants. Verifies that mission request and dispatch times are within a range of zero to 24 hours.

STORØ4

Stores contents of a type 04 card. Contents are the percentage of total requested missions for which to compute the maximum delay time. Also included are the duration (minutes) of the frequency interval for each mission priority. These are required to maintain counts of delayed missions.

STOR11

Stores contents of type 11 cards. Contents are scheduled mission requests by priority and time. Requests are encoded with their priority and time and stored in successive locations for use during the simulation. Normally at the beginning of each simulation day, the mission requests are loaded into the mission request queue.

STOR12

Stores the contents of a type 12 card -- the unscheduled mission request rates per time increment within priority.

STOR13

Forms a cumulative probability distribution of number of missions completed or delayed for each type of unscheduled mission request within priority.

ST1325

Edits and stores the contents of type 13 and type 25 cards -- the rate of occurrence of unscheduled mission requests. Card type 13 gives the rate for each mission type as a percentage of the total number of missions requested within its priority. Card type 25 gives the rate of occurrence as a daily frequency for each mission type within its priority. This routine also determines the highest priority and type for unscheduled mission requests.

STOR25

Forms a cumulative probability distribution for each type of unscheduled mission request within priority. Inputs are the daily frequencies for each type of unscheduled mission request by priority.

STOR43

Edits and stores the contents of type 43 cards. Contents are the number of steps of scheduled maintenance which each vehicle has completed.

STVHPF

Designed to determine and store the preferences for vehicle involvement in mission requests. These preferences are stored in an array indexed by mission and vehicle ID. If the mission ID or vehicle ID is out of the acceptable range of values, it is printed with an "illegal" message. Mission ID is then set to 1. In the case of an illegal vehicle ID, vehicle ID is set to the number of vehicle types.

TMINCR(I,J)

Converts clock time I (0001-2400) to time increments J (1-1440). For example, 0630 hours converts to time increment 78 when the time increment is five minutes.

TYPRATE

Determines move frequency for unit (IU) based on unit type.

VEHSUM(J)

Accumulates grand totals for all vehicles by summing individual vehicle type (J) totals. Accumulates total vehicle operating (travel) time, maintenance time, group forming time, and idle time.

VHMSPRF

Assigns vehicles to missions based on mission priority and vehicle preference. Prints table of computed vehicle assignment preference by unit mission and by mission type.

WRMSDL

Fills the array IOUT1(I,J) with mission type data and mission priority data and fills the array OUT1(I,J) with mission delay data. Both of these arrays are equivalenced and written in binary to a file NT2.

Chapter 4
MODEL USE

The TVFS model requires three basic types of input data: mission definition and demand information, vehicle fleet operational characteristics, and simulation control parameters. Table 4 describes the data required under each of these categories.

CURRENT INPUT REQUIREMENTS

All input data are specified on cards and are initially read onto four separate disc files by job control instructions. This procedure allows the input routines of the model itself to process the input data more efficiently.

The following table is designed to represent the sequencing of input data and helps to describe the logical files associated with each type of data.

Input Sequence Number	Type of Data	Logical File Name	Format Reference
1	Preplanned mission data	TAPE1	Table 5
2	Card type 52	TAPE3	Table 6
2.1	SRC unit data	TAPE3	Table 6
3	WES travel time data	TAPE4	Table 7
4	Parameter card data	TAPE5	Description of Parameter Cards

DESCRIPTION OF PARAMETER CARDS

All input data for the simulation is punched on 80-column cards in the format prescribed for the particular card type. A field consists of data from the beginning of a card to the first comma (first field) or blank (only field), the data between two commas (intermediate field), or the data from the last comma to the end of card or a blank (last field).

Table 4

TVFS INPUT DATA

Mission Related

- Unit destinations
- Kinds of missions and priority
- Mission tonnages
- Travel time to unit locations
- Allowable vehicle type(s) and preferences
- Load and unload factors

Vehicle Related

- Payload
- Operating speed
- Maintenance history
- Scheduled maintenance interval and duration
- Unscheduled maintenance interval and duration distributions

Operating Factors

- Simulation duration (days)
- Mission request start/finish
- Mission assignment start/finish
- Work day start/finish
- Mission discipline (cancel or day-to-day carryover)

Table 5

PREPLANNED MISSION DATA

Card Type		Description
11		Preplanned mission cards
⋮		⋮
11		
000		End of day
	or	
999		End of last day in cycle

Table 6

SRC UNIT DATA[a]

Card Column	Format	Description
1-9	A9	SRC number
10-38	29X	SRC unit designation (free field)
39-44	2A1,2I2	UTM[b] grid coordinates
45-50	F6.2	General cargo (tons)
51-56	F6.2	Ammunition resupply (tons)
57-62	F6.2	Bulk POL (tons)
63-68	F6.2	Retrieval/salvage (tons)
69-80	12X	Blank

[a]These data must be preceded by a type 52 card.

[b]Universal Transverse Mercator grid coordinates.

Table 7

WES TRAVEL TIME DATA[a]

Card Column	Format	Description
1-9	A9	SRC unit
10-14	F5.2	Distance
15-19	F5.2	Travel time for vehicle type 1
20-24	.	. 2
25-29	.	. 3
30-34	.	. 4
35-39	.	. 5
40-44	.	. 6
45-49	.	. 7
50-54	.	. 8
55-59	.	. 9
60-64	.	. 10
65-69	.	. 11
70-74	.	. 12
75-80	6X	Filler

[a]Parameter card type 27 (number of ammunition and cargo links) must agree with these data.

The first field of any card represents its type. It must be an integer. A card with a blank in column one is type zero. Data of some parameter cards are edited and stored by subroutines STORnn where nn is the type of the card.[1]

Card types and their general contents are presented in Table 8. On the succeeding pages of this chapter the card types are described in detail, including the contents of their fields and the corresponding program symbols.

---

[1]Data of other parameter cards are edited and stored by subroutines ISTOR1, ISTOR2, FSTOR1, FSTOR2, and CARDS.

Table 8

CARD TYPES AND CONTENTS

Card Type	Card Contents
"First"	Flexible output format
1	Modes of operation
2	Simulation times
3	Primes for random number generators
4	Frequency intervals for delayed missions
7	Removes scheduled missions
11	Scheduled and/or preplanned missions, by type
12	Unscheduled mission frequency by priority per time increment
13	Unscheduled mission frequency percentage (rate/day) by type and priority
14	Rewinds SRC file (NT3)
15	Vehicle preferences per mission type
18	Mission distances
19	Initiates call for reading of map coordinates (RDPOOL)
21	Supply point move indicators
22	Distances of pool supply point moves
25	Unscheduled mission frequency per day
27	Number of cargo/ammunition links
31	Vehicle maintenance hours per operating hour
32	Mean time between maintenance for vehicle types
34	Mission completion time intervals and priorities
41	Scheduled maintenance thresholds
42	Time until vehicle enters scheduled maintenance
43	Number of scheduled maintenance steps completed
44	Duration of scheduled maintenance
51	Mission payload (tons)
52	Group movement rates
53	Rate of FEBA advance (kilometers)
54	Grouping of missions for printing simulation results
55	SRC types per group

Table 8 (continued)

Card Type	Card Contents
65	Options for step-by-step listing of simulation results
66	Option to punch results of delayed missions
67	Initialize or reset number of SRC units available
71	Page heading title for listing of simulation results
72	Number of vehicles of each type
73	Vehicle speed in kilometers per hour
74	Vehicle turnaround times
75	Vehicle payloads (tons)
76	Vehicle operating costs
77	Vehicle overload factors
81	Number of loading docks
82	Loading rate (tons/minute)
83	Number of unloading docks
84	Unloading rate (tons/minute)
93	Initiates call for reading SRC data (RDSRCS)
94	Initiates call for reading vehicle travel times (RDTTM)
95	Option to print vehicle operating and maintenance characteristics
96	Lists input data
97	Begins simulation
98	Begins last simulation

Card Type Descriptions

FIRST CARD

Field	Contents	Program Symbol
1	Allows vehicle preferences if = 1, else restricts certain vehicle types to particular functions	NVHPRFØ
2	Prints every nth mission vehicle preference table	NDVHS
3	Format for printing vehicle preference	NFM9002

CARD TYPE 1

Field	Contents	Program Symbol
1	Card ID (=1)	N/A
2	Vehicle assignment option[a] (Ø or 1)	IEVEN
3	Procedures for servicing missions[b] (Ø or 1)	ICARRY
4	Type of daily report[c] (Ø, 1, 2, or 3)	DAOUSW
5	Road network factor	RDNWF
6	Request for preplanned missions ($\geq$1)	INPPM
7	Option to punch mission delay times[d] (Ø or 1)	MPUNCH
8	Highest mission priority; required if any preplanned missions have a higher priority than scheduled and stochastic missions	NUMPRS
9	Highest mission type; required if any preplanned missions have a higher type than scheduled and stochastic missions	N13
10	Vehicle times (vehicles/day)	NDPVA
11	Unit move maintenance factor	(USMMVF)

---

[a]Ø = Vehicles are assigned on a first available basis, i.e., vehicle #1 is always assigned first if available. 1 = Vehicles are assigned on a next available basis, i.e., the vehicle after the previously assigned vehicle is the next to be used if available.

[b]Ø = NON-CANCEL MODE -- unserviced missions are carried over to the next day. 1 = CANCEL MODE -- unserviced missions are cancelled at the end of the day.

[c]Ø = No daily report. 1 = Full report. 2 = Abbreviated report. 3 = Status report.

[d]Ø = Punch. 1 = No punch. Set = 1 assumes current fleet simulation. Set = Ø assumes pooled fleet simulation.

CARD TYPE 2

Field	Contents	Program Symbol
1	Card ID (=2)	N/A
2	Length of time increment in minutes	THPER
3	Time[a] vehicle assignments[b] begin	IASSB
4	Time vehicle assignments end	IASSE
5	Time work days begin	IDAYB
6	Time work days end	IDAYE
7	Time mission requests begin	IREQB
8	Time mission requests end	IREQE
9	Number of days to be simulated	LIMIT

[a] Time is specified in clock time, 0001-2400.

[b] Vehicle assignments imply servicing of mission requests.

CARD TYPE 3

Field	Contents	Program Symbol
1	Card ID (=3)	N/A
2	Quantity of numbers to be discarded from the random number generator (RANDØA) for unscheduled maintenance	NXRNUM
3	Quantity of numbers to be discarded from the random number generator (RANDØ) for the mission frequency	NXRNMQ
4	Quantity of numbers to be discarded from the random number generator (RAND2) for the mission types	NXRNMK
5	Quantity of numbers to be discarded from the random number generator (RAND1) for unit movement	NXRNMS
6	Quantity of numbers to be discarded from the random number generator (RAND3) for resupply request times	NXRNTM

CARD TYPE 4

Field	Contents	Program Symbol
1	Card ID (=4)	N/A
2	Percentage of missions (within each priority) for which to compute maximum delay time	CMPCTG
3	Frequency time interval (minutes) for delayed priority 1 missions	HSTMIN(1)
4	Frequency time interval (minutes) for delayed priority 2 missions	HSTMIN(2)
.		
.		
.		
N+2	Frequency time interval (minutes) for delayed priority N missions	HSTMIN(N)

CARD TYPE 7

Field	Contents	Program Symbol
1	Card ID (=7) [Removes scheduled missions]	N/A

CARD TYPE 11

Field	Contents	Program Symbol
1	Card ID (=11)	N/A
2	Mission priority	N/A
3	Time of mission (0001-2400)	N/A
4	First scheduled mission by type[a]	IDTREQ(I)[b]
5	Second scheduled mission by type	IDTREQ(I+1)
.		
.		
.		
N+3	Nth scheduled mission by type	IDTREQ(I+N-1)

[a]Value must be not greater than the number of missions for the simulation (field 9 of card type 1).

[b]I = sequential location of mission on card.

CARD TYPE 12

Field	Contents	Program Symbol
1	Card ID (=12)	N/A
2	Number of unscheduled priority 1 missions per time increment	FM(1)
3	Number of unscheduled priority 2 missions per time increment	FM(2)
.		
.		
.		
N+1	Number of unscheduled priority N missions per time increment	FM(N)

CARD TYPE 13

Field	Contents	Program Symbol
1	Card ID (=13)	N/A
2	Mission priority	J
3	Type for unscheduled mission in field 4	I
4	Frequency[a] of mission type I	SOR(I,J)
5	Frequency of mission type I+1	SOR(I+1,J)
.		
.		
.		
N+4	Frequency of mission type I+N	SOR(I+N,J)

[a]Frequency is expressed as a percentage (0-100) of the total for each priority.

CARD TYPE 14

Field	Contents	Program Symbol
1	Card ID (=14) [Rewinds SRC file (NT3)]	N/A

CARD TYPE 15

Field	Contents	Program Symbol
1	Card ID (=15)	N/A
2	Mission type	I
3	Vehicle type	N15
4	Mission preference for vehicle type N15	VEHPRF(I,N15)
5	Mission preference for vehicle type N15+1	VEHPRF(I,N15+1)
.		
.		
.		
N+3	Mission preference for vehicle type N15+N	VEHPRF(I,N15+N)

CARD TYPE 18

Field	Contents	Program Symbol
1	Card ID (=18)	N/A
2	SRC number	N18
3	Mission distance in kilometers for SRC N18	MSNDSKM(N18)
4	Mission distance in kilometers for SRC N18+1	MSNDSKM(N18+1)
.		
.		
.		
N+2	Mission distance in kilometers for SRC N18+N	MSNDSKM(N18+N)

CARD TYPE 19

Field	Contents	Program Symbol
1	Card ID (=19)	N/A

Initiates call of RDPOOL which reads card following card type 19, containing the following:

THIS CARD IS REQUIRED

Column	Contents	Program Symbol
1	First map code	MAP1
2	Second map code	MAP2
3-4	X-coordinate	POOLXP
5-6	Y-coordinate	POOLYP
7-12	Lag distance	POOLAG
13-18	Lead distance	POOLEAD
19-20	Number of priorities	NPLSRC
21-23	First priority	PRIORITY(1)
24-40	First mission name	MSNAME(1)
41-43	Second priority	PRIORITY(2)
44-60	Second mission name	MSNAME(2)
61-63	Third priority	PRIORITY(3)
64-80	Third mission name	MSNAME(3)

CARD TYPE 21

Field	Contents	Program Symbol
1	Card ID (=21)	N/A
2	Day of move	N21
3	Pool supply point move indicator[a] for day N21	MVPLSP(N21)
4	Pool supply point move indicator for day N21+1	MVPLSP(N21+1)
.		
.		
.		
N+2	Pool supply point move indicator for day N21+N	MVPLSP(N21+N)

[a]Ø = No move. 1 = Move.

CARD TYPE 22

Field	Contents	Program Symbol
1	Card ID (=22)	N/A
2	Day of move	N22
3	Distance of pool supply point move (kilometers) for day N22	DSPLSP(N22)
4	Distance of pool supply point move for day N22+1	DSPLSP(N22+1)
.		
.		
.		
N+2	Distance of pool supply point move for day N22+N	DSPLSP(N22+N)

CARD TYPE 25

Field	Contents	Program Symbol
1	Card ID (=25)	N/A
2	Mission priority	J
3	Mission type for field 4	I
4	Number of mission type I per day for this priority	SOR(I,J)
5	Number of mission type I+1 per day for this priority	SOR(I+1,J)
.		
.		
.		
N+4	Number of mission type I+N per day for this priority	SOR(I+N,J)

CARD TYPE 27

Field	Contents	Program Symbol
1	Card ID (=27)	N/A
2	Number of cargo links[a]	NCARGL
3	Number of ammunition links[a]	NAMMOL

[a] If > 5 or $\leq \emptyset$, default = $\emptyset$.

CARD TYPE 31

Field	Contents	Program Symbol
1	Card ID (=31)	N/A
2	Vehicle type	N31
3	Maintenance hours per operating hour for vehicle type N31	AMHFH(N31)
4	Maintenance hours per operating hour for vehicle type N31+1	AMHFH(N31+1)
.		
.		
.		
N+2	Maintenace hours per operating hour for vehicle type N31+N	AMHFH(N31+N)

CARD TYPE 32

Field	Contents	Program Symbol
1	Card ID (=32)	N/A
2	Vehicle type	N32
3	Mean time between maintenance for vehicle type N32	TBTM(N32)
4	Mean time between maintenance for vehicle type N32+1	TBTM(N32+1)
.		
.		
.		
N+2	Mean time between maintenance for vehicle type N32+N	TBTM(N32+N)

CARD TYPE 34

Field	Contents	Program Symbol
1	Card ID (=34)	N/A
2	Mission completion time interval (minutes)	MCTINT
3	Maximum number of mission priorities for which mission completion time is calculated	MCTMXM

CARD TYPE 41

Field	Contents	Program Symbol
1	Card ID (=41)	N/A
2	Vehicle type	N41
3	Scheduled maintenance threshold[a] (minutes) for vehicle type N41	SCHTRS(N41)
4	Scheduled maintenance threshold (minutes) for vehicle type N41+1	SCHTRS(N41+1)
.		
.		
.		
N+2	Scheduled maintenance threshold (minutes) for vehicle type N41+N	SCHTRS(N41+N)

[a]When a vehicle exits from unscheduled maintenance <u>and</u> the time remaining until its next <u>scheduled</u> maintenance is less than or equal to the threshold, then the vehicle enters scheduled maintenance.

CARD TYPE 42

Field	Contents	Program Symbol
1	Card ID (=42)	N/A
2	Scheduled maintenance step (1-16)	I
3	Vehicle type	J
4	Time (minutes) until vehicle type J enters step I of scheduled maintenance	HRSCH(I,J)
5	Time (minutes) until vehicle type J+1 enters step I of scheduled maintenance	HRSCH(I,J+1)
.		
.		
.		
2N+4	Time (minutes) until vehicle type J+N enters step I of scheduled maintenance	HRSCH(I,J+N)

CARD TYPE 43

Field	Contents	Program Symbol
1	Card ID (=43)	N/A
2	Vehicle type	J
3	Vehicle ID for field 4	I
4	Number of steps of scheduled maintenance that vehicle I of type J has completed	SCHIST(I,J)
5	Number of steps of scheduled maintenance that vehicle I+1 of type J has completed	SCHIST(I+1,J)
.		
.		
.		
N+4	Number of steps of scheduled maintenance that vehicle I+N of type J has completed	SCHIST(I+N,J)

CARD TYPE 44

Field	Contents	Program Symbol
1	Card ID (=44)	N/A
2	Scheduled maintenance step (1-16)	I
3	Vehicle type	J
4	Time (minutes) vehicle type J spends in step I of scheduled maintenance	DURSCH(I,J)
5	Time (minutes) vehicle type J+1 spends in step I of scheduled maintenance	DURSCH(I,J+1)
.		
.		
.		
2N+3	Time (minutes) vehicle type J+N spends in step I of scheduled maintenance	DURSCH(I,J+N)

CARD TYPE 51

Field	Contents	Program Symbol
1	Card ID (=51)	N/A
2	Mission number	N51
3	Payload (tons) for mission N51	MSNPYLD(N51)
4	Payload (tons) for mission N51+1	MSNPYLD(N51+1)
.		
.		
.		
N+2	Payload (tons) for mission N51+N	MSNPYLD(N51+N)

CARD TYPE 52

Field	Contents	Program Symbol
1	Card ID (=52)	N/A
2	Unit group	N52
3	Movement rate for group N52	MOVRATE(N52)
4	Movement rate for group N52+1	MOVRATE(N52+1)
.		
.		
.		
N+2	Movement rate for group N52+N	MOVRATE(N52+N)

CARD TYPE 53

Field	Contents	Program Symbol
1	Card ID (=53)	N/A
2	Simulation day	N53
3	Unit rate of advance for day N53	FEBAMVE(N53)
4	Unit rate of advance for day N51+1	FEBAMVE(N53+1)
.		
.		
.		
N+2	Unit rate of advance for day N53+N	FEBAMVE(N53+N)

CARD TYPE 54

Field	Contents	Program Symbol
1	Card ID (=54)	N/A
2	Mission number	M54
3	Group handling mission M54	MSNGRP(M54)
4	Group handling mission M54+1	MSNGRP(M54+1)
.		
.		
.		
N+2	Group handling mission M54+N	MSNGRP(M54+N)

CARD TYPE 55

Field	Contents	Program Symbol
1	Card ID (=55)	N/A
2	Group number	N/A
3	Group entry	N55
4	SRC for entry N55	TYPES(N55)
5	SRC for entry N55+1	TYPES(N55+1)
.		
.		
.		
N+3	SRC for entry N55+N	TYPES(N55+N)

CARD TYPE 65

Field	Contents	Program Symbol
1	Card ID (=65)	N/A
2-9	Options to print step-by-step simulation results (Ø = no print, 1 = print)	IPRNT1, IPRNT2, IPRNT3, IPRNT4, IPRNT5, IPRNT6, IPRNT7, IPRNT8

CARD TYPE 66

Field	Contents	Program Symbol
1	Card ID (=66)	N/A
2	Option to punch results of delayed missions (Ø = no punch, 1 = punch)	IPUNCH

CARD TYPE 67

Field	Contents	Program Symbol
1	Card ID (=67)	N/A
2	Initialize or reset number of available SRC units	NUNITS

CARD TYPE 71

Field	Contents	Program Symbol
1	Card ID (=71)	N/A
2	Page heading title for simulation results	TITLE(7Ø)

CARD TYPE 72

Field	Contents	Program Symbol
1	Card ID (=72)	N/A
2	Vehicle type	ITYPE
3	Number of ITYPE vehicles available	NUMVEH(ITYPE)
4	Number of ITYPE+1 vehicles available	NUMVEH(ITYPE+1)
.		
.		
.		
N+2	Number of ITYPE+N vehicles available	NUMVEH(ITYPE+N)

CARD TYPE 73

Field	Contents	Program Symbol
1	Card ID (=73)	N/A
2	Vehicle type	N73
3	Vehicle speed (KPH) for type N73	VHSPKPM(N73)
4	Vehicle speed (KPH) for type N73+1	VHSPKPM(N73+1)
.		
.		
.		
N+2	Vehicle speed (KPH) for type N73+N	VHSPKPM(N73+N)

CARD TYPE 74

Field	Contents	Program Symbol
1	Card ID (=74)	N/A
2	Vehicle type	N74
3	Vehicle turnaround time for type N74	VEHTAT(N74)
4	Vehicle turnaround time for type N74+1	VEHTAT(N74+1)
.		
.		
.		
N+2	Vehicle turnaround time for type N74+N	VEHTAT(N74+N)

CARD TYPE 75

Field	Contents	Program Symbol
1	Card ID (=75)	N/A
2	Vehicle type	N75
3	Payload (tons) for vehicle type N75	VHPYLD(N75)
4	Payload (tons) for vehicle type N75+1	VHPYLD(N75+1)
.		
.		
.		
N+2	Payload (tons) for vehicle type N75+N	VHPYLD(N75+N)

CARD TYPE 76

Field	Contents	Program Symbol
1	Card ID (=76)	N/A
2	Vehicle	N76
3	Operating cost for vehicle type N76	VHCOST(N76)
4	Operating cost for vehicle type N76+1	VHCOST(N76+1)
.		
.		
.		
N+2	Operating cost for vehicle type N76+N	VHCOST(N76+N)

CARD TYPE 77

Field	Contents	Program Symbol
1	Card ID (=77)	N/A
2	Vehicle type	N77
3	Overload percentage for vehicle type N77	VOLF(N77)
4	Overload percentage for vehicle type N77+1	VOLF(N77+1)
.		
.		
.		
N+2	Overload percentage for vehicle type N77+N	VOLF(N77+N)

CARD TYPE 81

Field	Contents	Program Symbol
1	Card ID (=81)	N/A
2	Mission number	N81
3	Number of loading docks for mission N81	MLDARAS(N81)
4	Number of loading docks for mission N81+1	MLDARAS(N81+1)
.		
.		
.		
N+2	Number of loading docks for mission N81+N	MLDARAS(N81+N)

CARD TYPE 82

Field	Contents	Program Symbol
1	Card ID (=82)	N/A
2	Mission number	N82
3	Loading rate (tons/minute) for mission N82	MSNLOAD(N82)
4	Loading rate for mission N82+1	MSNLOAD(N82+1)
.		
.		
.		
N+2	Loading rate for mission N82+N	MSNLOAD(N82+N)

CARD TYPE 83

Field	Contents	Program Symbol
1	Card ID (=83)	N/A
2	Mission number	N83
3	Number of unloading docks for mission N83	MULARAS(N83)
4	Number of unloading docks for mission N83+1	MULARAS(N83+1)
.		
.		
.		
N+2	Number of unloading docks for mission N83+N	MULARAS(N83+N)

CARD TYPE 84

Field	Contents	Program Symbol
1	Card ID (=84)	N/A
2	Mission number	N84
3	Unloading rate (tons/minute) for mission number N84	MSNUNLD(N84)
4	Unloading rate for mission number N84+1	MSNUNLD(N84+1)
.		
.		
.		
N+2	Unloading rate for mission number N84+N	MSNUNLD(N84+N)

CARD TYPE 93

Field	Contents	Program Symbol
1	Card ID (=93)	N/A

Invokes RDSRCS to read SRC data.

CARD TYPE 94

Field	Contents	Program Symbol
1	Card ID (=94)	N/A

Invokes RDTTM to read travel time data.

CARD TYPE 95

Field	Contents	Program Symbol
1	Card ID (=95)	N/A
2	Option to print ($\geq$1) workload, operating and maintenance requirements for each vehicle type (see MINVEHS, in program routines). Option to edit ($\geq$2) vehicle input characteristics.	N95

CARD TYPES 96, 97 AND 98

Field	Contents	Program Symbol
1	Card ID (=96) [Lists input data]	N/A
1	Card ID (=97) [Begins simulation]	N/A
1	Card ID (=98) [Begins last simulation]	N/A

INPUT DECK SET-UP

To illustrate the data input requirements and output options in specific detail, the deck set-up for the sample problem is presented in this section. The presentation is intended for the programmer-analyst who is responsible for assembling input data into formats and sequences accepted by the model. As such, this presentation primarily covers the manual procedures for setting up an input deck and for executing a simulation run. The meaning and functional rationale of the problem represented by the sample are fully discussed in Volume III of this documentation.

In a general sense, the TVFS model is capable of simulating several different kinds of real world vehicle operations provided those operations can be reduced to the general logical representations employed by the existing computer code. With only minor modification to this code, the scope of these capabilities can be increased significantly. This actual and potential capability is important to keep in mind when reviewing the following presentation of the sample problem which necessarily must be specific and, therefore, exclusive of other possible representations.

As shown in the previous section, the TVFS model requires several different kinds of input data. Some of these data are quite substantial in terms of the number of entries involved. Also, in many instances the sequence of data cards is critical to correct deck set up. The strictness of these requirements is partially offset, however, by flexible data card formats and by explicit designation of data type on most data cards. This latter feature permits automated identification of input data cards as they are read and alleviates some of the requirements for adhering to strict input card sequences.

The general aspects of deck set up discussed above will become clearer after examination of the detailed sample problem. Appendix C lists the input deck, including job control cards, for the sample problem. A summary of the various input formats and related data elements is also provided in Appendix C. Selected parts of the resultant output are shown in Appendix D. Also, results of other runs are shown of the same problem, with the only difference among runs being the print options that

were selected. Comparison of these results indicates the wide variability in the kinds of output that are available via the print options.

DATA PREPARATION

The previous sections concerning data input requirements and corresponding data formats describe the necessary information needed to organize data for a simulation run, with the exception of the WES travel time data (TAPE4). These data are supplied by WES and are converted by a separate computer program, WESEXT (WES Extract), to the form described in the previous sections.

Purpose

The purpose of this program is to produce, on tape, a file of records, each of which consists of a JOBSRC, a distance to be travelled, and travel times for each of 11 vehicle types.

Input

WESEXT requires five types of input records. Two of these are input via card file; three are on the input tape.

The first card in the card file contains the program parameters which indicate the weather option and the number of scenario snapshots to be extracted for each route-type weather-type combination. (See Appendix E.)

The remainder of the card file consists of mission records, each of which contains an SRC number, an origin identifier, and a destination identifier. These mission records are grouped together to constitute all of the missions for a particular snapshot set. A blank card follows the last mission of every snapshot set as a delimiter. (See Appendix E.)

The first record on the WES input tape consists of three values: the first indicates the number of node pairs in the succeeding job directory of origin-destination identifiers; the second indicates the number of sets of vehicle records succeeding the job directory; the third indicates the total number of records of all types contained on the tape, including this first record. The first record also contains an alphanumeric heading indicating the scenario theater and the date of the creation of the data tape. (See Appendix E.)

The second type of record on the input tape consists of 20 origin-destination identifiers. There will be as many of this type of record as is indicated by the second value of the first record, as discussed above. The entire set of this second type of record consistutes a job directory. (See Appendix E.)

The third type of record on the input tape consists of an origin identifier; a destination identifier; a vehicle type; a distance (from origin to destination); and vehicle travel times under dry weather, wet weather, and snow or sand conditions for route types 1, 2, and 3. (See Figure 4.) This type of record is grouped into sets containing a record for each of 17 vehicle types. There will be as many of these sets as is indicated by the first value of the first record, as described earlier.

Output

WESEXT generates two categories of output. One category is printed; the other is written to tape. Since tape output is intended to be used to punch cards, this category of output will be referred to as one which is "punched." An annotated listing of printed output is presented in Appendix E.

Printed output, for the purpose of verifying the correctness of both input and output, consists of a listing containing the following items:

1. Mapping sequence of vehicle types. This line of vehicle types indicates the order by which the WES data are rearranged with regard to vehicle types. This rearrangement is undertaken to decrease execution time of the TVFS model by grouping vehicles of similar degrees of mobility so that search time in locating vehicles with relatively equal mobility characteristics is less. Table 9 indicates the TVFS vehicle type reordering with regard to WES data.

2. The first record from the input tape.

3. Job directory records from the input tape.

4. A snapshot number and one set of mission records from the card file.

5. A table indicating, first, the number of missions in the current snapshot, followed by all of the job directory indices which are to be used in filling the appropriate weather-route snapshot matrix with travel times for each of the vehicle types.

Items 4 and 5 are listed for as many scenario snapshots as are indicated by the program parameter card.

6. The first five sets of the third type of input from the tape.

7. A heading indicating a snapshot number, route type, and weather type.

DESCRIPTION	ORIGIN NODE	DESTIN. NODE	VEH. TYPE NO.	DIST. (X1ØØ) MILES	WEATHER TYPE								
					1 - DRY			2 - WET			3 - SNOW OR SAND		
					RTE 1 TIME	RTE 2 TIME	RTE 3 TIME	RTE 1 TIME	RTE 2 TIME	RTE 3 TIME	RTE 1 TIME	RTE 2 TIME	RTE 3 TIME
FORMAT	I6	I6	I8	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø	I1Ø

Figure 4 —Organization of Travel Times Record

Table 9

MAPPING

TVFS Index	WES Index	Vehicle Type ID Corresponding to WES Index
1	7	M151
2	8	M715
3	9	M35
4	11	M813
5	14	M125
6	15	M818
7	1	M561
8	2	M656
9	3	M520
10	6	M548
11	16	TW.D.W.
12	10	M49
13	13	M816
14	4	M559
15	5	M553
16	12	M821
17	17	M60A

8. A listing of the missions, mission distances, and travel times for the current snapshot. This list corresponds to the data which are "punched."

Items 7 and 8 are printed for each snapshot and route type combination for one weather type. For each consecutive weather type, the printed output is repeated, beginning with item 6.

"Punched" output consists of card image records to be written on tape. The first of these records pertains to ammunition and cargo common links. These may be identified by a hyphen appearing in position 5 of the record. In the case of common link records, the first positions of the record contain a common link identifier, and the next four positions contain a destination identifier. Remaining records of this type not pertaining to common links differ in that the first five positions contain the first five characters of an SRC. After the first nine positions of a record are a value for distance and nine travel times, one for each route-type weather-type combination. The order of these is as described earlier with regard to the third type of input record. As many of these records per snapshot will be produced as card input records per snapshot are read into the program. This may occur for several route-type weather-type combinations. The "punched" output tape can later be assigned to a card punch to produce a deck for input to the TVFS model.

Processing

Extraction of WES data from the input tape and card file takes place as shown in Figure 5. The job directory is read from the tape and stored in an array. The parameter card, containing the maximums for weather types and snapshots, is read. One snapshot set of records is read from the card file and stored in an array. Each of these records contains an SRC number and its origin and destination identifiers. This array is stepped through a record at a time. A match on origin-destination identifiers is sought in the job directory array and, when found, the relative address of the job directory is stored in an index array. After the snapshot array is completely stepped through in this manner, the vehicle travel times data for the snapshot of interest are read from the input tape and stored in an array. Then, using the index array to locate the appropriate travel times for unique origin-destination sets of interest, WESEXT

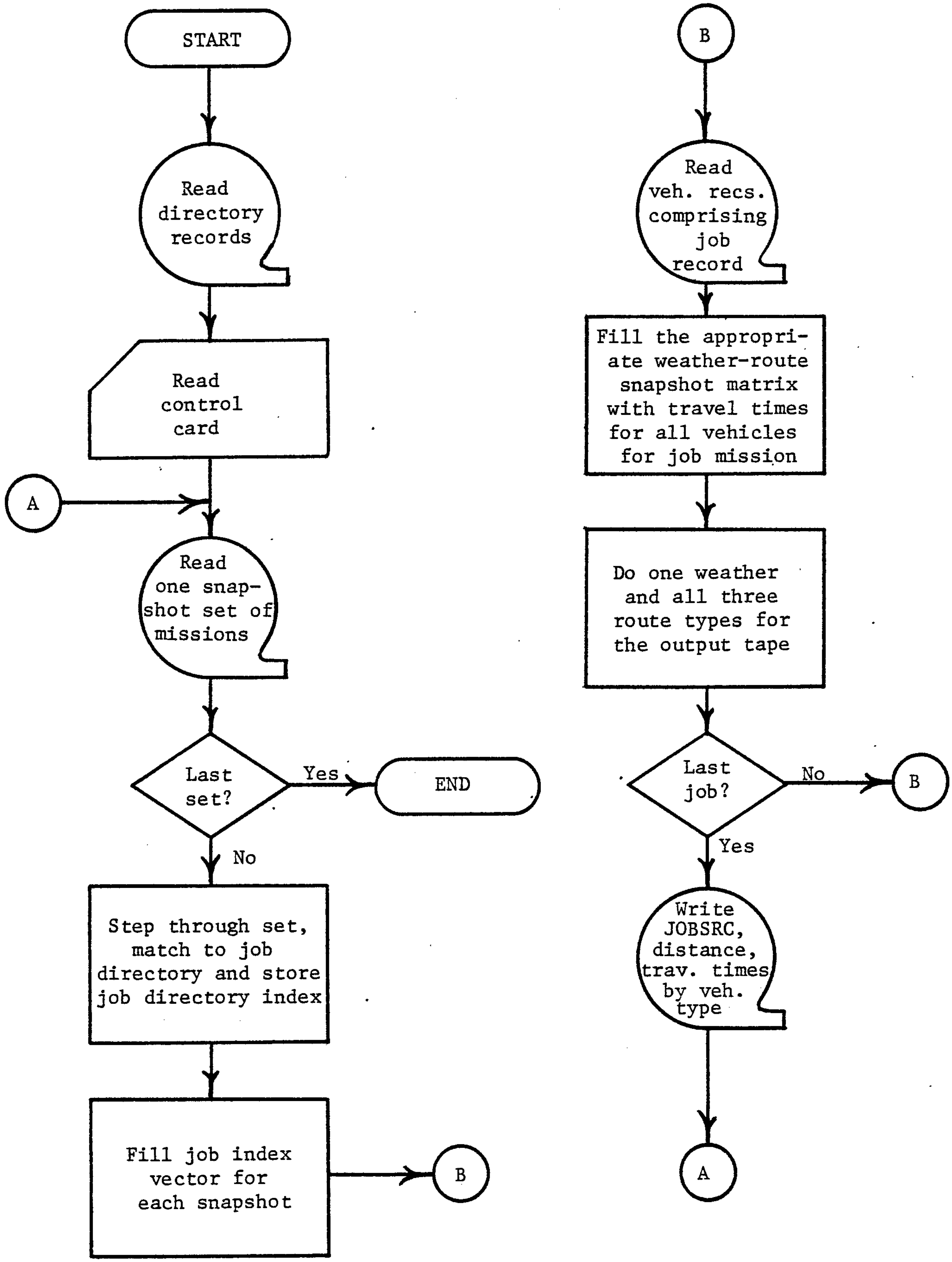

Figure 5—WESEXT Flow Diagram

extracts the proper times for each route-type weather-type combination under consideration when a match is found for each snapshot set. This procedure develops a weather-route snapshot matrix with travel times for all vehicle types for the current snapshot. These data are written on tape for later punching, and are also printed for verification purposes.

JOB CONTROL PROCEDURES

The model is designed to run on the CDC 6000/7000 series computer, using the FORTRAN extended compiler (FTN). The source code of the model is structured as a program library tape using the CDC UPDATE processor. UPDATE provides a convenient means of maintaining source decks in a compressed, addressable format. That format is referred to as a program library tape file and is initially established from source decks via a creation run through UPDATE. Once established via UPDATE, the program library tape facilitates alteration, compilation and execution of source programs with a minimum of system instructions, peripheral devices, and user input. Figure 6 lists the job control instructions and deck setup for a typical compilation and execution run.

As with any computer program, the time and resources required to compile the program can be greater than the resources required to execute it. In the case of this model, where many runs may be made without modification to the source code, it is advisable to execute the program from a binary file, as opposed to recompiling the model each time it is executed. Figure 7 lists the job control instructions and deck setup to create a binary file using an UPDATE program library. Figure 8 lists the instructions required to execute this binary file.

The model has an average compile time of 64 seconds on a CDC 6600, using approximately 60K of core. Any cross referencing requested at compile time will add additional time for this job step. The R=3, FTN parameter, which generates cross reference tables of variables and indices, increases the compile time to 85 seconds.

In the event that modifications to the model are necessary, UPDATE provides instructions for inserting, deleting or replacing program instructions. For a description of these instructions, the reader should consult the CDC UPDATE reference manual. There are also system functions available which allow subroutines to be compiled individually after changes are made, rather than recompiling the entire model. The COPYL option allows this procedure and can be found in the Scope reference manual.

```
job card
REQUEST,OLDPL,NORING,VSN=TVFS.
UPDATE(F,INPUT=DUMMY)
FTN(A,I=COMPILE)
COPYBF(INPUT,TAPE1)
COPYBF(INPUT,TAPE3)
COPYBF(INPUT,TAPE4)
REWIND(TAPE1,TAPE3,TAPE4)
LGO.
REWIND(TAPE7)                   Required if delayed missions times to be punched.
COPYBF(TAPE7,PUNCH)
REWIND(TAPE8,TAPE9,TAPE10)
COPYBF(TAPE8,PUNCH)             Required if delayed missions times by priority
COPYBF(TAPE9,PUNCH)             to be punched.
COPYBF(TAPE10,PUNCH)
7/8/9
Preplanned mission data
7/8/9
SRC unit data
7/8/9
WES travel time data
7/8/9
Parameter card input data
6/7/8/9
```

Figure 6—Compilation and Execution via UPDATE

```
job card
REQUEST, OLDPL,NORING,VSN=TVFS.
REQUEST,LGO,RING,VSN=BINARY.
UPDATE(F,INPUT=DUMMY)
FTN(A,I=COMPILE)
REWIND(LGO)
6/7/8/9
```

Figure 7—Generation of a Binary Program Library

```
job card
REQUEST,LGO,NORING,VSN=BINARY.
COPYBF(INPUT,TAPE1)
COPYBF(INPUT,TAPE3)
COPYBF(INPUT,TAPE4)
REWIND(TAPE1,TAPE3,TAPE4)
LGO.
7/8/9
Preplanned mission data
7/8/9
SRC unit data
7/8/9
WES travel time data
7/8/9
Parameter card input data
6/7/8/9
```

Figure 8—Execution of a Binary Program Library

Chapter 5
MODEL OUTPUTS

The TVFS model is designed to give a comprehensive listing of results for each simulation day. These results include such information as basic input data, data error messages, daily activity of mission completion and mission delay times, vehicle assignment and other time-related information. While these data may be sufficient for many applications, additional print options are available to the user for more detailed output results. All optional output is controlled by input card types 1, 65, 66 and 95.

OUTPUT CONTROL OPTIONS

Field number 4 of input card type 1 can be set to one of four values (0, 1, 2 or 3) based on the user's desire for no daily report, a full daily report, an abbreviated report, or simply a daily status report, respectively.

Input card type 65 allows the setting of eight print options. Each of these options invokes intermediate output results from various subroutines. Table 10 lists the print options available and gives an overview of what each option controls. Note that within a selected subroutine a particular print option may invoke printing of data different from that invoked by another print option.

Card type 95 controls the printing of workload, operating and maintenance requirements for each vehicle type. This card provides a dual function in that if it is set to 1, it will trigger the additional printer output; and if set to 2, it will invoke editing of vehicle input characteristics, as well as provide the additional printed output.

The model also provides options for punching the results of delayed missions. If set to 1, card type 66 will generate a file (TAPE7) of delayed mission times as well as provide additional print information displaying

Table 10

PRINT OPTION LISTING (CARD TYPE 65)

Card Field	Variable Name	Subroutine and Associated Data
1	N/A	(=65)
2	IPRNT1	ASMNT1: Mission request and mission delay time. Delay times, to date, for each mission type and priority. Cumulative mission delay times/priority. Total missions completed/priority. Total vehicles waiting to form groups. ASMNT2: Maintenance time/sortie. Vehicle number of each type now being used to fulfill request. ASMNT3: Excess vehicles of type j. Number of vehicles in each mission group for current requests. Vehicle type j loading/unloading rates. INSERT: Inserts or deletes next available vehicle of type j to current mission request.
3	IPRNT2	ASMNT1B: Scheduled and unscheduled maintenance time/ sortie. Vehicle/mission turnaround time. Operating (travel) time/sortie. Vehicle number of type j now being used to fulfill requests. Next time vehicle i of type j is available. Soonest available vehicle of type i on next day. FSTOR1, FSTOR2, ISTOR1, ISTOR2: Number of vehicles of type j read into the simulation. MOVSRC: Data pertaining to unit moves, to include new X and Y coordinates, date of move, number of unit moves, payload and new distance from pool. PRNTIN: For each priority mission, prints number of unscheduled missions.
4	IPRNT3	ASMNT3: Number of vehicles of type j now being used to fulfill requests. Type vehicles needed for current mission request. Next time vehicle i of type j is available. Vehicle preference for current mission. Number of vehicles of type j waiting. Group completion count for type j vehicles. Priority of mission request now being serviced. CALCDST: Prints distance of unit from supply point.
5	IPRNT4	ASMNT2: Number of times vehicle i of type j has entered scheduled maintenance. Vehicle number of type j now being used to fulfill request. Number of scheduled maintenance steps this run. Scheduled maintenance time/ sortie.

Table 10 (continued)

Card Field	Variable Name	Subroutine and Associated Data
		CALCDST: Distance of unit from pool.
		DELETE: Location of priority j requests in queue. Time and day of this mission request. Next available location in mission queue. Last location in mission request queue.
		NXTREQ: Prints generated list of mission requests for priority j for next day.
		QUESCH: Prints type of unscheduled mission to be put in queue, and time of event. Insertion place in queue for next scheduled or preplanned mission.
		SRTPER: Prints cumulative probability for mission type i of priority j.
6	IPRNT5	Not used in this application of the model.
7	IPRNT6	IGPSRV: Number of type j vehicles required for a type i mission. Sortie group value. Vehicle type. Mission payload (tons).
8	IPRNT7	DSPTCHR: Time of day.
		MNSPCT: Mission frequency and payload (tons). Number and percentage of sorties per day. Distance (km) one way. Vehicle speed (kph). Sortie time (hrs). Vehicle overload factor. Vehicle payload (tons).
		MNTVEH: Average sortie time (hrs). Probability of no maintenance for a vehicle of type j. Probability of maintenance for a type j vehicle. Average maintenance time. Expected maintenance time given maintenance occurs for a type j vehicle.
		OUTPT3: Longest unscheduled maintenance time.
		RDPOOL: Prints pool map coordinates along with lead and lag distances (km).
		RDSRCS: Prints SRC type along with X and Y map coordinates. Number of days since last move.
9	IPRNT8	NEWSRCS: Lists SRC type along with X and Y map coordinates. Number of days since last unit move.

these times. Field number 7 of card type 1 has a dual function in that if set to 1, it assumes the model is operating with the current fleet and writes to file 8 (TAPE8), file 9 (TAPE9), or file 10 (TAPE10) depending on the mission priority. If this field is set to zero, the model assumes a pooled fleet simulation and no output files are written. The pooled fleet option is not exercised in the most current version of the model (i.e., HIMO version).

SAMPLE REPORTS

The TVFS model follows the same systematic method of providing its output as the system flow described in Chapter 3. The model steps through a series of input routines reading, validating and storing the contents of the input files. Upon successfully processing the input data, the model generates the following report topics:

- Current program maximums list
- Random number generator discard information
- WES travel time/vehicle type matrix
- Verification of input parameters
- SRC unit location tableau
- Vehicle preference table
- Vehicle travel and turnaround time report
- Scheduled maintenance interval and duration report

The current program maximums list describes the physical and logical boundaries of model parameters. These boundaries are programming considerations and cannot be exceeded without dimension changes within the model's code.

The list of random numbers discarded from the number generators is an image of input card type 3 and reflects the first N numbers discarded.

The WES travel time data represent the links upon which missions may travel, giving the distances, travel time by vehicle type, and common cargo and ammunition links.

The verification of input data report lists basic timing and daily clock information such as mission requests begin and end times, work day begin and end times, and time interval between clock increments. The report also lists basic information concerned with whether mission requests are carried over from day to day or whether missions not completed at the end of each day are dropped.

A table is generated which, for each SRC type, gives the distance the unit is from the common links, its coordinates, and the number of loading and unloading docks by mission types.

The vehicle preference table lists, by mission, the preference for a vehicle of a certain type. A 1 indicates that a vehicle of that type may be allowed to perform the mission, while a Ø indicates that a vehicle of that type cannot.

The report dealing with vehicle one-way travel and turnaround times lists the TAT and vehicle speed for each vehicle type by SRC unit. Time and distance of common links are not included in this report. The report also lists the cost, payload, speed and maintenance history of each of the vehicle types in the model.

A maintenance table for each step of scheduled maintenance is listed in report format such that the duration and interval of each vehicle type at each maintenance step is displayed. Following this report is another maintenance history schedule displaying the vehicle number under each vehicle type, representing the number of maintenance steps each vehicle of a vehicle type has completed.

If particular input data options are selected, some of the above reports may not appear or may appear in more detailed format -- beyond what results as standard output reports.

The model's output routines are designed to address subject areas and report topics sequentially rather than duplicating reports for each day of simulation. Interruption of standard reports is primarily a function of print options selected by the user. The sequence of simulation results is listed in Appendix D. Due to the large and varied amount of information displayed, the reader is encouraged to study these reports and follow the output routines from mission request to mission completion. Within missions, the reader will note the vehicle assignment network and scheduled maintenance tableau.

There are reports displaying mission numbers within mission priority; number of cargo, ammunition and unit moves; group completion rates; mission group forming times; mission tonnages; overload tonnages; and an array of intermediate reports triggered by various print options, simulated events and end of day conditions.

SENSITIVITY OF RESULTS

The outputs of the TVFS model can be significantly affected both by the selection of control parameter values and by the specification of input variable values. Consistent with the orientation of this volume to programmer considerations, sensitivity of results to the former type of inputs is discussed here. Sensitivity of substantive analytic results to input variable values is discussed in Volume III.

In order to control the types of reports generated by the model, to assess the correct functioning of the model, to verify data set-up, and to evaluate trade-offs between the size of simulation runs, the programmer must be familiar with output generation capabilities of the model. This includes a detailed understanding of how input control parameters, default values, override values, and even input variable values affect the kind and amount of output reports. To a limited extent, these relationships can be documented in summary form; more extensive description is prohibitive because of the large, indefinite number of combinations deriving from all the different potential problem applications of the model.

Amount and Type of Output

There are several specific input data types which are dominant factors in determining the length of the model's output. An understanding of these inputs can be helpful in estimating both the execution time of the model and the amount of model output anticipated. Table 11 represents a general summarization of the model's input-output relationships.

If the user elects to set up restarts or multiple cases within the same runstream, many of the routines used to initialize, read and validate the input data are bypassed. This feature is advantageous in that it eliminates excessive input data listings and the model only has to deal with data which are altered from one case to the next. Table 12 shows the approximate amount of output produced by each output routine if, in fact, that routine is exercised by input data organization.

Table 11

SUMMARY OF MODEL INPUT-OUTPUT RELATIONSHIPS

Affected Report	Input Data Type
WES travel time matrix	Number of SRC units Number of vehicles Number of common links
SRC unit location table	Number of SRC units Number of mission types
Vehicle preference table	Number of vehicle types Number of SRC units Number of mission types
Vehicle specification report	Number of vehicle types
Scheduled maintenance table	Number of vehicle types Number of scheduled maintenance steps
Scheduled maintenance history table	Number of vehicle types Number of vehicles per type
Mission completion table	Number of days of simulation Number of vehicles Number of mission priorities
Computer vehicle assignment preference table	Number of vehicle types Number of vehicles per type Number of missions

Table 12

OUTPUT PAGE ESTIMATES

Output Routine/Report	Report Size
SRC units location tableau	50 units per page
Vehicle preference table	55 missions per page
Vehicle travel time report	50 units per page
Scheduled maintenance interval and duration report	18 maintenance steps per page
Scheduled maintenance history by vehicle types	N = number of vehicle types times number of vehicles per type (N/400 pages)
Mission completion table	50 missions times NDVHS[a]
Computed vehicle assignment preference table	50 missions times NDVHS[a]
Daily simulated mission results	3 simulated days per page

[a]Refer to definition of parameter cards (FIRST card)
NDVHS = 1, prints all missions.
NDVHS = 2, prints every other mission.
NDVHS = 3, prints every third mission.
NDVHS = X, prints every X mission.